2013

Algorithms for constructing more accurate and inclusive phylogenetic trees

Ruchi Chaudhary
Iowa State University

Algorithms for constructing more accurate and inclusive phylogenetic trees

by

Ruchi Chaudhary

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Science

Program of Study Committee:

David Fernández-Baca, Major Professor

Srinivas Aluru

Oliver Eulenstein

Ryan Martin

Giora Slutzki

Iowa State University

Ames, Iowa

2013

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

First and foremost, I thank my advisor David Fernández-Baca for his excellent guidance, boundless patience, and immense support throughout this research. His focus on the quality of research, and the delicate balance between challenge and support always reenergized me. One of the most important things I have learnt from him is writing scientific research clearly and concisely. I hope my future research articles will not disappoint him!

I would also like to thank my committee members: Srinivas Aluru, Maria Axenovich (who served in my committee until last few months), Oliver Eulenstein, Ryan Martin (who agreed to serve in my committee at the last moment), and Giora Slutzki, for being encouraging and appreciative of my work. I would additionally like to thank Oliver for his guidance on some of the projects we did together. Many thanks to J. Gordon Burleigh for helping me in conducting empirical research systematically, providing me biological data sets, and answering my endless questions. His guidance has been extraordinary in understanding the biological impact of my theoretical results throughout my research.

I want to thank David Fernández-Baca and Pavan Aduri for teaching me two of the best computer science courses: *Design and Analysis of Algorithms* and *Theory of Computation*, respectively, in the most effective manner.

I am indebted to my former and current lab mates Mukul Bansal, Wen-Chieh Chang, Akshay Deepak, Jianrong Dong, Bradley Shutters, Sudheer Vakati, and André Wehe for being such a good friends and collaborators. Akshay's discordant path in each of our discussion topics made me acknowledge the diversity of convictions. André gave some laughs to the otherwise serious lab. Additional thanks to Akshay, André, and Wen-Chieh for helping in resolving hardware and software issues of my lab computer.

Thanks to Abigail Andrews, Darlene Brace, Maria-Nera Davis, Linda Dutton, Cindy Marquardt, and Laurel Tweed for always being so helpful, approachable, and friendly.

Finally, I want to thank my family, especially my husband Sumit for encouraging me to start PhD when I was so lost back in 2008. Not only have you enabled me to do this research, but you also have kept my spirits high in all ups and downs throughout this thesis. I thank my most wonderful parents for their love, affection, and support: my dad for realizing me the importance of higher education and my mom for giving me freedom to pursue what I wanted to. Thanks to my brother and sister for their love and support over the years. A special thanks to my mom-in-law for helping me continue my research when I needed it the most, and my dad-in-law for always being so appreciative. Last but not least, I want to thank my little son Manav for being such a epitome of patience. Thanks to all his teachers at day care for giving him excellent care while I was in school.

Ruchi Chaudhary

ABSTRACT

Despite the unprecedented outpouring of molecular sequence data in phylogenetics, the current understanding of the tree of life is still incomplete. The widespread applications of phylogenies, ranging from drug design to biodiversity conservation, repeatedly remind us of the need for more accurate and inclusive phylogenies. My thesis addresses some of the underlying challenges, by presenting theoretical and empirical results, as well as algorithms for a range of phylogenetic optimization problems.

In the first part of this thesis, I develop a heuristic method for the NP-hard unrooted Robinson-Foulds (RF) supertree problem, and show that it yields more accurate supertrees than those obtained from Matrix Representation with Parsimony (MRP) and rooted RF heuristic. In the second, I present an RF distance measure based approach (MulRF) for inferring a species tree from the input multi-copy gene trees, through a generalization of RF distance to multi-labeled trees. Through simulation, I show that this approach, which is independent of gene tree discordance mechanisms, produces more accurate species trees than existing methods when incongruence is caused by gene tree error, duplications and losses, and/or lateral gene transfer. Next, I perform a simulation study to evaluate the performance of Gene Tree Parsimony (GTP) under duplication and duplication and loss cost models and compare it to MulRF method. The objective is to study the effects of various types of sampling (e.g., gene tree and sequence sampling), gene tree error, and duplication and loss rates on the accuracy of the phylogenetic estimates by GTP and MulRF. Next, I present efficient error correction algorithms for gene tree reconciliation based on duplication, duplication and loss, and deep coalescence. In the end, I present NP-completeness proofs for two problems whose complexity was previously unknown.

CHAPTER 1. Introduction

Life is ubiquitous throughout the nature—organisms spreading from the poles to the equator and from the bottom of the sea to several miles above the ground, and surviving from the freezing to well over boiling water temperatures. A confirmation to that is 1.75 million biological species that have been identified and described to date, and yet it is believed to be a fraction of the total diversity on earth. Despite that, the best evidences strongly suggest that all life on earth has a common ancestor. Since that common ancestor appeared more than 3.5 billion years ago, ancestral species have split to form new and independent species (*speciation*), with their own physical manifestation and genetic makeup. Rather rarely, some of these otherwise independent species have also come together to form yet another species (*hybridization*) or to exchange genetic material (*lateral gene transfer*).

Phylogenetics is the exploration and identification of the evolutionary relationships among the many different kinds of species on earth, both living (*extant*) and dead (*extinct*). These evolutionary relationships are frequently represented by a branching tree, called a *phylogenetic tree* or a *phylogeny* or just a *tree*. Occasionally, when reticulation events such as hybridization, lateral gene transfer, or recombination are believed to be involved, the *phylogenetic networks* are also considered to represent evolutionary relationships. They differ from phylogenetic trees by the addition of nodes with two parents (*hybrid nodes*), instead of nodes with only one parent (*tree nodes*). The prime focus of this thesis is restricted to the phylogenetic trees and their problems.

Methods for building phylogenetic trees follow one of the two underlying approaches or philosophies: *phenetic*, considers only the similarities and differences of character data (e.g. physical traits or molecular sequences), and *cladistic*, considers the molecular sequences as well as various possible pathways of evolution that brought to these sequences. The ultimate goal

is to construct the phylogenetic hypothesis of all species on earth, the so-called *Tree of Life*.

Phylogenies have widespread applications. They are important for tracking the evolution of diseases, and thus help design drugs and vaccines (for example the development of influenza vaccine (Bush et al., 1999)). Plant scientists use phylogenies to determine the genes associated with positive traits such as the ability to survive adverse growing conditions (e.g., drought). The collective knowledge can certainly be applied towards breeding more productive crops, and feeding more people, as a result. Phylogenies are used in Biogeography to hypothesize the biological distribution of organisms (Lomolino et al., 2005), results are also applicable to biodiversity conservation decision-making (Erwin, 1991). Other applications comprise protein structure prediction and multiple-sequence alignment. The importance of phylogenies in biology presses the need to have more inclusive and accurate phylogenies. This thesis addresses several of such theoretical, computational was well as experimental problems that comes in way towards reaching that goal.

The proliferation of next generation sequencing technologies has revolutionized phylogenetics by incorporating large genomic data sets into phylogenetic inference. However, it also has drawn attention to complex patterns of genomic variation that result from processes such as gene duplication and loss, incomplete lineage sorting, recombination, or lateral gene transfer (e.g., Maddison (1997)). These processes can create conflict among gene tree topologies and obscure or mislead phylogenetic analyses (e.g., Mossel and Vigoda (2005); Kubatko and Degnan (2007); Beiko et al. (2008); Penny et al. (2008)). Thus, in order to accurately hypothesize the phylogenetic relationships from genomic data, it is necessary to address the incongruence among input gene trees. Furthermore, a preferred method for such phylogenetic analyses not only has to address the incongruence among input trees but also remain computationally tractable for constructing more comprehensive phylogenies. Chapters 3 - 7, develop fast, both asymptotically and in practice, algorithms and study existing techniques that allows analysis on such conflicting input gene trees. We start with Chapter 2, that details the preliminaries for this thesis. See Appendix A for commonly used symbols in each chapter.

In Chapter 3, *Fast Local Search for Unrooted Robinson-Foulds Supertrees*, we address the supertree problem, which has multiple applications in systematics. Supertrees combine mul-

tiple, usually conflicting, species trees with partially overlapping taxon sets into phylogenies, containing all species from the input trees. We describe the unrooted Robinson-Foulds (RF) supertree problem that also allows non-binary input trees. This problem is NP-hard, thus a heuristic method is required to estimate solutions for large data sets. Our heuristics are based on the Edge Contract and Refine (ECR) operation. Experiments on simulated and empirical data sets show that our method yields better supertrees than those obtained from Matrix Representation with Parsimony (MRP) and rooted RF heuristic. These results have been published in (Chaudhary et al., 2012b).

A *multi-labeled tree*, or *mul-tree* in short, is a tree in which multiple leaves can have the same label. MUL-trees are omnipresent in the literature under different names such as tangled trees (used in host-parasite co-evolution (Page, 1994)), area cladograms (used in Biogeography, i.e., the study of geographical distribution of organisms (Lomolino et al., 2005)), and multi-copy gene trees (i.e., a gene tree containing more then one sequence from a species). The ability to use mul-trees as input, instead of being restricted to single copy genes, allows a phylogenetic method to incorporate the wealth of genomic data from multi-copy genes, not only single-copy genes, into phylogenetic inference. Our next contribution, is a new technique that also allows multi-copy gene trees in phylogenetic analysis.

In Chapter 4, *Inferring Species Trees from Incongruent Multi-Copy Gene Trees Using the Robinson-Foulds Distance*, we present a new tree distance metric based approach for inferring species trees from incongruent multi-copy gene trees. Unlike most previous methods, this approach does not assume that gene tree incongruence is caused by a single, specific biological process or gene tree error. Consequently, it is appealing for analyses of genomic data sets, in which many unknown biological processes as well as phylogenetic errors likely contribute to the conflicts between the gene trees and the species tree. By generalizing RF distance measure to multi-labeled trees, we formulate the MulRF problem which seeks a species tree that minimizes the total RF distance to the input multi-copy gene trees. We present a fast heuristic algorithm for the MulRF problem, and compared its performance with multiple gene tree parsimony (GTP) approaches using gene tree simulations that incorporate gene duplications and losses and/or lateral transfer. We found that the MulRF method produces more accurate species

trees than various gene tree parsimony approaches, emphasizing that a phylogenetic method based on a generic tree distance metric may be more appropriate if the conflict among genes is due to error and/or to multiple, interacting biological processes. Furthermore, the MulRF heuristic runs quickly on data sets containing hundreds of trees with up to a hundred taxa, showing its suitability for large-scale phylogenomic analyses. A paper containing these results is under review.

In Chapter 5, *A Simulation Study to Compare Two Non-parametric Approaches for Species Trees Construction*, we study which phylogenetic approach among GTP and MulRF can best resolve a species tree from multi-copy genes. We use gene simulations to evaluate the performance of GTP under duplication and duplication and loss cost models and compare them to the mechanism-free MulRF method. We look at the effects of various samplings (e.g., gene tree and sequence sampling), gene tree error, and duplication and loss rates on the accuracy of the phylogenetic estimates by Only-dup, Dup-loss, and MulRF. As expected, the species trees were more accurately estimated for increased gene tree and gene sequence sampling and decreased duplication and loss rate by all three methods. Further, the error in the gene trees negatively affected the species tree analyses. In general, Only-dup performed poorly compared to the other two methods, MulRF was best in estimating small species trees ($\leq$ 100 taxa) and Dup-loss larger species trees ($\geq$ 250 taxa). Our results also highlight the limitations of Only-dup in estimating duplications and losses, and Dup-loss in losses. A manuscript of these results is under construction.

Chapter 6, *Efficient Error Correction Algorithms for Gene Tree - Species Tree Reconciliation*, deals with the error in the gene trees in a complete different way for improved gene tree - species tree reconciliation, and more accurate species tree construction, as a result. Gene tree - species tree reconciliation problems infer the patterns and processes of gene evolution within a species tree. GTP approaches seek the evolutionary scenario that implies the fewest gene duplications, duplications and losses, or deep coalescence events needed to reconcile a gene tree and a species tree. While a GTP approach can be informative about genome evolution and phylogenetics, error in gene trees can profoundly bias the results. We introduce efficient algorithms to correct gene tree topologies based on the gene duplication, duplication and loss,

or deep coalescence cost models. The algorithms work by identifying the small rearrangements, based on *Tree Bisection and Reconnection* and *Subtree Prune and Regrafting* tree operations on the gene trees, that reduce the reconciliation cost. They are extremely fast and thus amenable to analyses of large-scale genomic data sets. The results have been published in Chaudhary et al. (2012a)

In Chapter 7, *NP-Completeness Proofs*, we present NP-completeness proofs for two problems whose complexity was previously unknown. The first problem is computing the RF distance between two multi-labeled trees. This question was originally introduced by Ganapathy et al. (2006) as the problem of computing RF distance between two area cladograms in Biogeography, but as multi-labeled trees are omnipresent (see for example, Chapter 4), this problem also appeals to different areas of research for various applications. The second problem is a tree labeling problem: Labeling two unlabeled trees so as to minimize the RF distance between the resulting singly-labeled trees. The reduction of it is partially similar to the first problem, but its applications are in the study of tree shapes. The first proof here is undergoing review process and the manuscript for the second is under construction.

In Chapter 8, we summarize our results and give some concluding remarks.

CHAPTER 2. Preliminaries

2.1 Phylogenetic Trees

An *unrooted phylogenetic tree* is an acyclic connected graph, with no vertices of degree two and every leaf (vertex of degree one) labelled uniquely. Internal vertices (vertices that are not leaves) are typically left unlabelled. We use "phylogenetic tree" and "tree" interchangeably.

A *rooted phylogenetic tree* is defined in the same way except one internal vertex, which can have degree two, is distinguished as *root*. The remaining internal vertices have degree three or more.

In a *binary unrooted tree* every internal vertex has degree three. A *binary rooted tree* has every internal vertex of degree three, except the root which has degree two.

One example of a binary tree is a *caterpillar tree*. An unrooted caterpillar tree has one central path with leaves branching off it (see Fig. 2.2(1)). In an rooted caterpillar tree, leaves append to a single path from the root to the single leaf (see Fig. 2.2(2))

Let the leaf set of a (rooted or unrooted) tree T be denoted by $\mathcal{L}(T)$. The set of all vertices

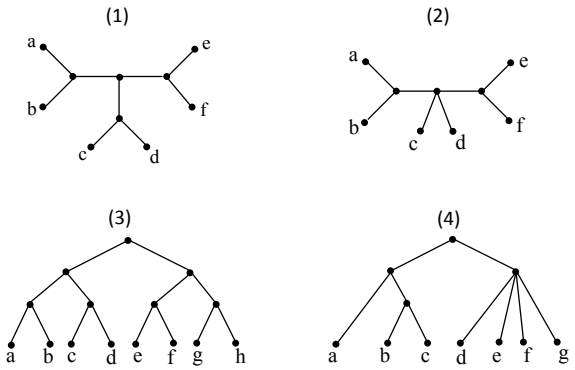

Figure 2.1 Example of phylogenetic unrooted and rooted trees. Tree in, (1) is binary, unrooted, (2) is non-binary, unrooted, (3) is binary, rooted, and (4) is non-binary, rooted.

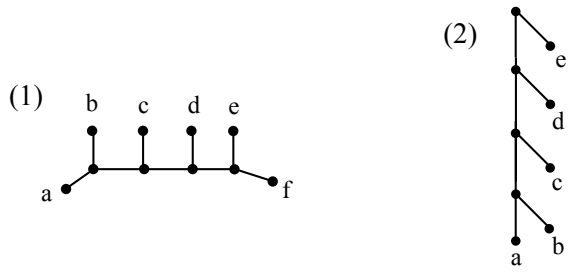

Figure 2.2 (1) Unrooted caterpillar tree, (2) rooted caterpillar tree.

by $V(T)$ and the set of all edges by $E(T)$. The root of a rooted tree T is denoted by $rt(T)$

Given a rooted tree T, a vertex v is *internal* if $v \in V(T) \backslash (\mathcal{L}(T) \cup rt(T))$. The set of all internal vertices of T is denoted by $I(T)$. We define $\preceq_T$ to be the partial order on $V(T)$ where $x \preceq_T y$ if y is a vertex on the path from $rt(T)$ to x. If $\{x, y\} \in E(T)$ and $x \preceq_T y$, then y is the *parent* of x and x is a *child* of y. Two vertices in T are *siblings* if they have the same parent.

Let T be the given rooted tree. The *least common ancestor (LCA)* of a non-empty subset $L \subseteq V(T)$, denoted by $\mathrm{LCA}_T(L)$, is the unique smallest upper bound of L under $\preceq_T$.

Let T be a (rooted or unrooted) tree and U be a subset of $V(T)$. We denote by $T(U)$ the minimum subtree of T that connects the elements in U. The *restriction* of unrooted tree T to U, denoted by $T_{|U}$, is the phylogenetic tree that is obtained from $T(U)$ by suppressing all vertices of degree two. In restricting rooted tree T to U, all non-root vertices of $T(U)$ are suppressed.

2.1.1 Split and Clusters

Let T be a unrooted tree and e be an edge of T. Removal of e subdivides T into two components. Let A be the set of leaves in one component and B be the set of leaves in another component. A and B are called the *parts* of the resulting *split* $A|B$. Order does not matter, so $A|B$ is identical to $B|A$. Each edge in a tree induces a unique split. A split is called *nontrivial* if each of A and B contains at least two elements. The set of all nontrivial splits of a tree T, denoted by $\Sigma(T)$, is called a *split set of* T.

Now consider a rooted tree T. Let v be a vertex in T. The subtree of T rooted at vertex $v \in V(T)$, denoted by T_v, is the tree induced by $\{u \in V(T) : u \preceq v\}$. For each node $v \in I(T)$,

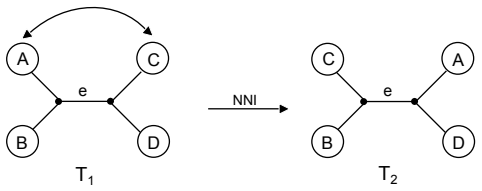

Figure 2.3 An NNI operation. Tree T_2 results from T_1 after swapping subtree A with C.

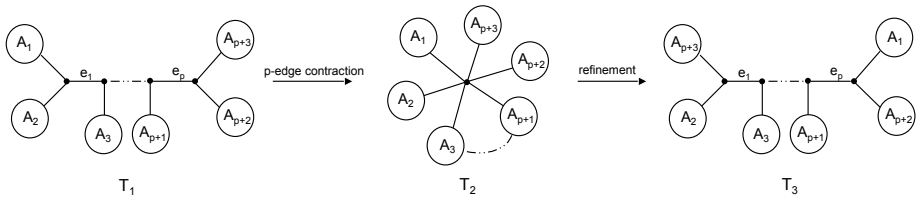

Figure 2.4 A p-ECR operation. Tree T_2 results from T_1 from contracting edges $e_1, e_2, ..., e_p$; T_3 is a full refinement of T_2. Observe the degree $p+3$ vertex in T_2. Note that, in general, the edges contracted do not have to be adjacent, as they are in this example.

$C_T(v)$ is defined to be the set of all leaf nodes in T_v. Set $C_T(v)$ is called a *cluster*. The set of all clusters of a tree T, denoted by $\mathcal{H}(T)$, is called a *cluster set of* T.

2.2 The Tree Edit Operations

2.2.1 Nearest Neighbor Interchange (NNI) Operation

Let T_1 be an unrooted, binary tree and let e be an internal edge of T_1. An *NNI operation* on T_1 consists of swapping one of the two subtrees on one side of e with one of the two subtrees on the other side of e (Allen and Steel, 2001). (See Fig. 2.3.)

2.2.2 p-Edge Contract and Refine (ECR) Operation

Let T be an unrooted, binary binary tree. A *p-ECR operation* on T is the result of (i) choosing p internal edges $e_1, e_2, ..., e_p$ of T, (ii) contracting $e_1, e_2, ..., e_p$ and (iii) constructing some full refinement of the resulting tree (Ganapathy et al., 2003). (See Fig. 2.4.) Note that the 1-ECR operation is equivalent to NNI operation.

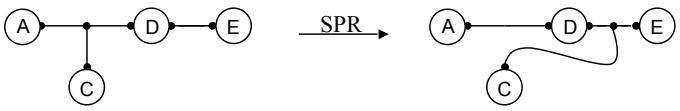

Figure 2.5 An SPR operation. Subtree C is cut and regrafted to a new vertex between subtree D and E.

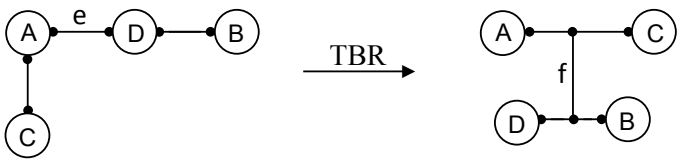

Figure 2.6 An TBR operation. Edge e is cut and an new edge f is added between the two components.

2.2.3 Subtree Prune and Regraft (SPR) Operation

An SPR operation on an unrooted, binary tree T cuts any edge, thereby pruning a subtree t, and then regrafts t by the same cut edge to a new vertex obtained by subdividing a pre-existing edge in $T - t$ (Allen and Steel (2001); Bordewich and Semple (2004)). (See Fig. 2.5.)

2.2.4 Tree Bisection and Reconnection (TBR) Operation

An TBR operation on an unrooted, binary tree T cuts an edge e, and then adds a new edge f between a vertex that subdivides an edge of one component of $T\backslash e$ and a vertex that subdivides an edge of the other component of $T\backslash e$ (Allen and Steel, 2001). (See Fig. 2.6.)

2.2.5 Contraction and Refinement Operation

Let T be a rooted or unrooted tree. The *contraction* of an edge in T collapses that edge and identifies its two endpoints. The *refinement* of an unresolved vertex (i.e., an internal vertex with degree greater than three) expands that vertex into two vertices connected by an edge. Contraction and refinement can be viewed as inverses of each other (Fig. 2.7).

2.3 Multi-labeled Trees

An *unrooted phylogenetic mul-tree* or *unrooted mul-tree*, is a tuple $\mathcal{T} = (T, M, \varphi)$ consisting of an unrooted tree T, called *underlying* tree, a set of labels M, and a surjective *labeling function*

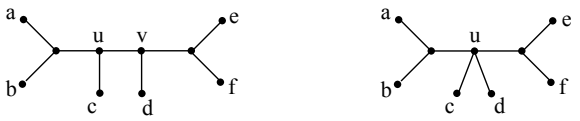

Figure 2.7 The contraction of edge $\{u, v\}$ in the first tree produces the second tree; conversely, the refinement of vertex u in the second tree produces the first tree.

$\varphi : \mathcal{L}(T) \to M$ that maps each leaf of T with a label in M. A mul-trees in which each leaf has the same label (i.e., $|M| = 1$) is called a *uniform mul-tree*.

Informally, a mul-tree is simply an unrooted phylogeny in which multiple leaves can have the same label. For any label $\ell \in M$, $\varphi^{-1}(\ell)$ is the set of all leaves labeled ℓ. If φ is a bijection, the corresponding unrooted mul-tree is just a (singly-labeled) unrooted tree.

Similarly, a *rooted phylogenetic mul-tree* or *rooted mul-tree*, is a tuple $\mathcal{T} = (T, M, \varphi)$ consisting of an rooted tree T, a set of labels M, and a surjective *labeling function* $\varphi : \mathcal{L}(T) \to M$ that maps each leaf of T with a label in M. Note that the difference between unrooted and rooted mul-trees is in the underlying tree which is a rooted or unrooted tree for rooted and unrooted mul-trees, respectively. In this thesis, we use the traditional notation for a tree when the given mul-tree is clearly a tree.

The concepts introduced above for unrooted trees naturally extend to mul-trees. For example, a mul-tree $\mathcal{T} = (T, M, \varphi)$ is binary if T is binary. Two mul-trees $\mathcal{T}_1 = (T_1, M, \varphi_1)$ and $\mathcal{T}_2 = (T_2, M, \varphi_2)$ are isomorphic if T_1 and T_2 are isomorphic under bijection $\tau : V(T_1) \to V(T_2)$ such that $\varphi_1(u) = \varphi_2(\tau(u))$ for all $u \in \mathcal{L}(T_1)$.

2.4 Robinson-Foulds Distance

The *Robinson-Foulds (RF)* distance between two mul-trees (or trees) T_1 and T_2, denoted by $RF(T_1, T_2)$, is the minimum number of contractions and refinements necessary to transform T_1 into a mul-tree (or tree) isomorphic to T_2 (Robinson and Foulds, 1981).

In case of unrooted trees, the RF distance can be equivalently defined via split set. For two unrooted trees T_1 and T_2 (Robinson and Foulds, 1981),

$$RF(T_1, T_2) := |(\Sigma(T_1) \backslash \Sigma(T_2)) \cup (\Sigma(T_2) \backslash \Sigma(T_1))|.$$

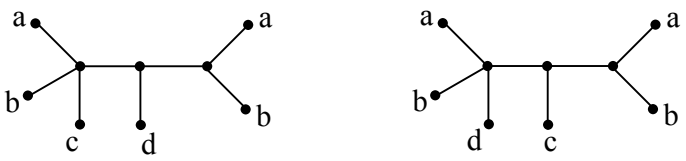

Figure 2.8 Two mul-trees that induce the same set of splits but are not isomorphic.

For rooted trees, the RF distance can be equivalently defined via cluster set. For two rooted trees T_1 and T_2 (Robinson and Foulds, 1981),

$$RF(T, S) := |(\mathcal{H}(T_1) \backslash \mathcal{H}(T_2)) \cup (\mathcal{H}(T_2) \backslash \mathcal{H}(T_1))|.$$

Unlike trees, it is possible for two unrooted (or rooted) mul-trees T_1 and T_2 to satisfy $\Sigma(T_1) = \Sigma(T_2)$ (or $\mathcal{H}(T_1) = \mathcal{H}(T_2)$) and yet not be isomorphic (see Fig. 2.8). Thus, the RF distance between two mul-trees cannot be computed by splits or clusters. In fact, we prove that computing the RF distance between two mul-trees is NP-complete (Chapter 7).

CHAPTER 3. Fast Local Search for Unrooted Robinson-Foulds Supertrees

3.1 Introduction

Supertree techniques are widely used to combine multiple, usually conflicting, species trees with partially overlapping taxon sets into phylogenies (supertrees) containing all species from the input trees (Bininda-Emonds et al., 2007; Davies et al., 2004; Pisani et al., 2002). Matrix representation with parsimony (MRP) (Baum, 1992; Ragan, 1992) is by far the most commonly used supertree method. While MRP often performs well (Bininda-Emonds and Sanderson, 2001; Chen et al., 2006; Eulenstein et al., 2004), MRP supertrees may display size and shape biases, and contain relationships that are not supported by any of the input trees (Goloboff, 2005; Purvis, 1995; Pisani and Wilkinson, 2002). Still, MRP remains popular because it can take advantage of fast and effective parsimony heuristics and incorporate a broad range of input data, including rooted, unrooted, and non-binary trees (Bininda-Emonds et al., 2005).

In contrast to MRP, the Robinson-Foulds (RF) supertree method seeks a supertree that minimizes the total RF distance to the input phylogenies (Bansal et al., 2010b). Thus, an RF supertree is consistent with the maximum number of splits in the input trees. Although the properties of the RF supertree method make it a desirable alternative to MRP, its use has been limited by the lack of effective heuristics. Bansal et al. (Bansal et al., 2010b) recently developed fast local search algorithms for the *rooted* RF problem, the special case where the input trees and the supertree are rooted. Here, we describe new local search algorithms for the *unrooted* RF problem. These are not only asymptotically as fast as the rooted RF heuristics, but they also allow more types of input data and improve the quality of supertree estimates, making the RF supertree method a viable alternative to MRP for nearly any data set.

The use of local search (hill-climbing) heuristics for constructing RF supertrees is motivated

by the NP-hardness of the underlying optimization problem. Local searches explore the space of possible supertrees in search of a *locally optimum* supertree, a tree whose score is minimum within its "neighborhood", where the neighborhood is defined by a *tree edit operation*. The best known tree edit operations are Nearest Neighbor Interchange (NNI) (Allen and Steel, 2001), Subtree Prune and Regraft (SPR) (Allen and Steel, 2001; Bordewich and Semple, 2004), and Tree Bisection and Reconnection (TBR) (Allen and Steel, 2001). The sizes of the respective neighborhoods are $\Theta(n)$, $\Theta(n^2)$, and $\Theta(n^3)$, where n is the number of taxa in the tree.

Ganapathy et al. introduced the p-Edge Contract and Refine (ECR) operation (Ganapathy et al., 2003), which is based on selecting a set of p edges to contract, after which all possible refinements of the contracted tree are generated. While the 1-ECR operation is equivalent to NNI, for larger values of p, p-ECR allows us to explore tree space in ways that other operations do not. In particular, although the size of the TBR neighborhood is big (i.e., $\Theta(n^3)$), the number of 2-ECR neighbors that are also TBR neighbors is just $O(n)$ (Ganapathy et al., 2003, 2004). Thus, a 2-ECR search can cover a significant part of the tree space left unexplored by TBR search. The effectiveness of combining TBR with ECR has been demonstrated for parsimony (Goloboff, 1999). Further, the RF-distance between two trees is at most $2p$ if and only if they are one p-ECR move apart (Ganapathy et al., 2004). This suggests that ECR may be particularly well-suited for building RF supertrees.

We present fast NNI and 2-ECR local search algorithms for the unrooted RF supertree problem. We also discuss how to extend these results to handle p-ECR, for any fixed p. In particular, our algorithms perform a $O(n)$ time preprocessing step after that RF distance from each tree in NNI and 2-ECR neighborhood can be computed in constant time. To our knowledge, the only previous related work is (Bansal et al., 2010b) and the supertree analysis package Clann (Creevey and McInerney, 2005), which implements a heuristic for maximizing the number of splits shared between the input trees and the supertree, but lacks any running time performance guarantees. Our NNI and 2-ECR search algorithms run in $\Theta(nk)$ and $\Theta(n^2k)$ time, where k is the number of input trees. They represent $\Theta(n)$ speed-ups over the naïve solutions for these problems. The algorithms produce binary supertrees, but the input trees are not required to be binary. The techniques we use are, on the surface, similar to those

used earlier for rooted trees (Bansal et al., 2010b). In particular, we transform the unrooted problem into a rooted one and use an LCA mapping technique related to that of (Bansal et al., 2010b). On the other hand, there are some important differences. For unrooted trees, we use LCA mappings from the supertree to each input tree, the opposite of what is done for rooted trees. This simplifies the algorithm considerably and allows us to compute RF distances without restricting the supertree to the leaf set of each input tree. It also enables us to handle multiple alternative rootings cleanly.

We examine the performance of our unrooted ECR-based RF supertree heuristic using several large simulated and empirical data sets, and compare its performance with rooted RF supertrees obtained by SPR-based local search (Bansal et al., 2010b). We demonstrate that the ability to handle unrooted trees allows us to construct, in a reasonable amount of time, higher-quality trees than those obtained by assuming fixed roots.

The results presented here are not only of algorithmic interest. It is often beneficial, if not necessary, to allow unrooted input in supertree analyses. Identifying the root of a species tree is among the most difficult problems in phylogenetics (e.g., Smith (1994); Wheeler (1990); Sanderson and Shaffer (2002)), and conventional likelihood and parsimony-based phylogenetic methods typically produce unrooted trees. To root trees, most analyses include outgroup taxa that lie outside the clade of interest. However, in many cases, no useful outgroups exist, or the phylogenetic distance of available outgroups may contribute to systematic, or "long-branch attraction", errors (Wheeler, 1990; Leebens-Mack et al., 2005). Methods for rooting trees in the absence of an outgroup also can be problematic. For example, rooting the tree by assuming a molecular clock, or similarly using mid-point rooting, may be misled by molecular rate variation throughout the tree (Holland et al., 2003; Huelsenbeck et al., 2002), and the use of non-reversible models appears to perform well only when the substitution process is strongly asymmetric (Huelsenbeck et al., 2002; Yap and Speed, 2005).

3.2 Unrooted RF Supertree Problem

A *profile* is a tuple of trees $\mathcal{P} := (T_1, T_2, ..., T_k)$, where each tree $T_i \in \mathcal{P}$ is called an *input tree*. A *supertree* on $\mathcal{P}$ is a phylogenetic tree S such that $\mathcal{L}(S) = \bigcup_{i=1}^{k} \mathcal{L}(T_i)$. We write n to

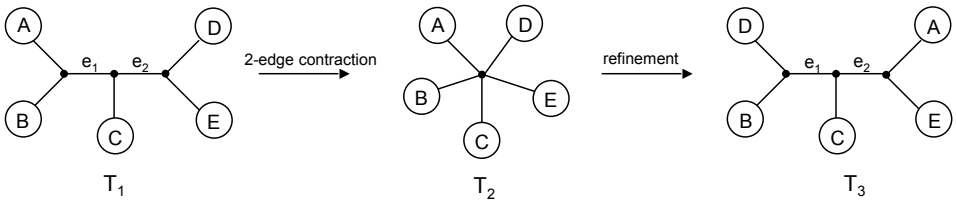

Figure 3.1 A 2-ECR operation. Tree T_2 results from T_1 after contracting edge e_1 and e_2; T_3 is a full refinement of T_2. Observe the degree five vertex in T_2.

denote $|\mathcal{L}(S)|$; i.e., n is the total number of distinct leaves in the profile.

We extend the notion of RF distance to profile and supertree as follows. Let $\mathcal{P}$ be a profile of unrooted trees and S be a supertree for $\mathcal{P}$. Then, the *RF distance* from $\mathcal{P}$ to S is $RF(\mathcal{P}, S) := \sum_{T \in \mathcal{P}} RF(T, S)$.

We now state our main problem. Let $\mathcal{B}(\mathcal{P})$ be the set of all binary supertrees for $\mathcal{P}$.

Problem 1 (**Unrooted RF Supertree**)**.**

Input: A profile $\mathcal{P} = (T_1, T_2, ..., T_k)$ of unrooted trees.

Output: A supertree S^* for $\mathcal{P}$ such that $RF(\mathcal{P}, S^*) = \min_{S \in \mathcal{B}(\mathcal{P})} RF(\mathcal{P}, S)$.

The unrooted RF supertree problem is NP-hard even when all input trees have the same leaf set (McMorris and Steel, 1993). In the rest of this Chapter, we develop local search heuristics for the unrooted RF supertree problem based on the p-ECR tree edit operation. Our primary focus here is on the special cases of the p-ECR operation where $p = 1$ and $p = 2$. See Figs. 2.3 and 3.1. Note that the 1-ECR operation is equivalent to the well-known *Nearest Neighbor Interchange* (NNI) operation.

Let p-ECR$_T$ denote the set of trees that can be obtained from a binary tree T by applying a single p-ECR operation. The *p-ECR search problem* is defined as follows:

Problem 2 (p-ECR Search)**.**

Input: A profile $\mathcal{P} = (T_1, T_2, ..., T_k)$ of unrooted trees and a binary supertree S for $\mathcal{P}$.

Output: A tree $S^* \in p\text{-ECR}_S$ such that $RF(\mathcal{P}, S^*) = \min_{S' \in p\text{-ECR}_S} RF(\mathcal{P}, S')$.

We give algorithms that solve the 1-ECR/NNI and 2-ECR search problems in time $\Theta(nk)$ and $\Theta(n^2k)$, respectively. We achieve this by first executing a $O(nk)$-time preprocessing step (explained in Section 3.3), which is the same for both problems. After that, for each tree in

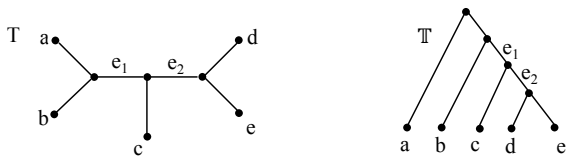

Figure 3.2 Tree T with leaf set $\{a, b, c, d, e\}$. The rooted tree $\mathbb{T}$ with $r = a$ is also shown.

the input profile, the RF distance from any tree in 1-ECR$_S$ or 2-ECR$_S$ can be computed in constant time. After explaining these algorithms, we outline an extension to p-ECR for any fixed p.

3.3 Preprocessing

3.3.1 The Connection to Rooted RF Distance

We now show how to compute the RF distance from an arbitrary input tree T to a supertree S by working with rooted versions of these trees.

Suppose S is a supertree for a profile $\mathcal{P}$ of unrooted trees and let T be a tree in $\mathcal{P}$. Throughout the rest of this Chapter, we assume that some arbitrary but fixed taxon $r \in \mathcal{L}(T) \cap \mathcal{L}(S)$ is chosen for T. We refer to r as the *outgroup*. Different outgroups may be used for different input trees. Let $\mathbb{T}$ and $\mathbb{S}$ be the trees that result from rooting T and S at the respective branches incident on r (see Fig. 3.2).

Lemma 1. *Let T and S be two unrooted phylogenetic trees with $\mathcal{L}(T) = \mathcal{L}(S)$, then,*

$$RF(T, S) = RF(\mathbb{T}, \mathbb{S}).$$

Proof. We first show that $RF(T, S) \leq RF(\mathbb{T}, \mathbb{S})$. Recall that $RF(T, S) = |(\Sigma(T) \backslash \Sigma(S)) \cup (\Sigma(S) \backslash \Sigma(T))|$. We prove that for each unmatched split in the split set of T (respectively, S), there exists a unique unmatched cluster in the corresponding rooted tree $\mathbb{T}$ (respectively, $\mathbb{S}$). Let $A|B$ be a split such that $A|B \in \Sigma(T)$ but $A|B \notin \Sigma(S)$. Assume without loss of generality that $r \in A$. Then, $B \in \mathcal{H}(\mathbb{T})$ but $B \notin \mathcal{H}(\mathbb{S})$. The argument for S and $\mathbb{S}$ follows similarly. Thus $RF(T, S) \leq RF(\mathbb{T}, \mathbb{S})$ holds. The proof that $RF(T, S) \geq RF(\mathbb{T}, \mathbb{S})$ is similar. □

We extend RF distance to the case where $\mathcal{L}(\mathbb{T}) \subseteq \mathcal{L}(\mathbb{S})$ in the same way as for unrooted trees. That is, $RF(\mathbb{T}, \mathbb{S}) := RF(\mathbb{T}, \mathbb{S}_{|\mathcal{L}(\mathbb{T})})$, where $\mathbb{S}_{|\mathcal{L}(\mathbb{T})}$ is the rooted phylogenetic tree obtained from $\mathbb{S}(\mathcal{L}(\mathbb{T}))$ by suppressing all non-root vertices of degree two. We now show how to compute the RF distance in this more general setting, without explicitly building $\mathbb{S}_{|\mathcal{L}(\mathbb{T})}$.

Definition 1 (Restricted Cluster). Let $v \in I(\mathbb{S})$. The *restriction* of $C_{\mathbb{S}}(v)$ to $\mathcal{L}(\mathbb{T})$ is defined as

$$\hat{C}_{\mathbb{T}}(v) := \{w \in \mathcal{L}(\mathbb{S}_v) : w \in \mathcal{L}(\mathbb{T})\}.$$

$\hat{C}_{\mathbb{T}}(v)$ is called a *restricted cluster*.

Definition 2 (Vertex Function). The *vertex function* $f_{\mathbb{S}}$ assigns each $u \in I(\mathbb{T})$ the value $f_{\mathbb{S}}(u) = |U|$, where $U := \{v \in I(\mathbb{S}) : C_{\mathbb{T}}(u) = \hat{C}_{\mathbb{T}}(v)\}$.

Fig. 3.3 shows the tree $\mathbb{T}$ after labeling the internal vertices with the values of the vertex function. Two vertices in $\mathbb{T}$ have label 1 since each one of them has exactly one identical restricted cluster in $\mathbb{S}$. (The LCA mapping depicted in Fig. 3.3 is explained later.) Observe that if $\mathcal{L}(\mathbb{S}) = \mathcal{L}(\mathbb{T})$, then for all $u \in I(\mathbb{T})$, $f_{\mathbb{S}}(u) \leq 1$.

We use $f_{\mathbb{S}}$ to define the following set, which is used to compute $RF(\mathbb{T}, \mathbb{S})$.

$$\mathcal{F}_{\mathbb{S}} = \{u \in I(\mathbb{T}) : f_{\mathbb{S}}(u) = 0\}$$

We drop the subscript from $f_{\mathbb{S}}$ and $\mathcal{F}_{\mathbb{S}}$ when it is clear from the context.

Lemma 2. *Let* $\mathbb{S}' := \mathbb{S}_{|\mathcal{L}(\mathbb{T})}$. *Then* $RF(\mathbb{T}, \mathbb{S}) = |I(\mathbb{S}')| - |I(\mathbb{T})| + 2|\mathcal{F}_{\mathbb{S}'}|$.

Proof. Recall that $RF(\mathbb{T}, \mathbb{S}) = |(\mathcal{H}(\mathbb{T}) \backslash \mathcal{H}(\mathbb{S}')) \cup (\mathcal{H}(\mathbb{S}') \backslash \mathcal{H}(\mathbb{T}))|$. Let $\mathcal{G}_{\mathbb{S}'}$ be a set $\{u \in I(\mathbb{T}) : f_{\mathbb{S}'}(u) > 0\}$. Thus, $RF(\mathbb{T}, \mathbb{S}) = |I(\mathbb{S}')| + |I(\mathbb{T})| - 2|\mathcal{G}_{\mathbb{S}'}|$. Since $|\mathcal{G}_{\mathbb{S}'}| + |\mathcal{F}_{\mathbb{S}'}| = I(\mathbb{T})$ we have $RF(\mathbb{T}, \mathbb{S}) = |I(\mathbb{S}')| - |I(\mathbb{T})| + 2|\mathcal{F}_{\mathbb{S}'}|$. □

Lemma 3. *Let* $\mathbb{S}' := \mathbb{S}_{|\mathcal{L}(\mathbb{T})}$. *Then* $|\mathcal{F}_{\mathbb{S}}| = |\mathcal{F}_{\mathbb{S}'}|$.

Proof. We prove the lemma by showing that for $u \in I(\mathbb{T})$, $f_{\mathbb{S}}(u) \neq 0$ if and only if $f_{\mathbb{S}'}(u) \neq 0$. ($\Rightarrow$) Since $f_{\mathbb{S}}(u) \neq 0$, there exists a vertex v in $\mathbb{S}$ such that $C_{\mathbb{T}}(u) = \hat{C}_{\mathbb{T}}(v)$. There are two cases.

Case 1: $\mathcal{L}(\mathbb{S}_v) = \hat{C}_{\mathbb{T}}(v)$. In this case v stays in S' after restriction, and so $f_{\mathbb{S}'}(u) \neq 0$.

Case 2: $\hat{C}_{\mathbb{T}}(v) \subset \mathcal{L}(\mathbb{S}_v)$. Let the children of v be v_1 and v_2. If $\mathcal{L}(\mathbb{T})$ is not disjoint with $\mathcal{L}(\mathbb{S}_{v_1})$ and $\mathcal{L}(\mathbb{S}_{v_2})$, then v exists in $\mathbb{S}'$. Otherwise, at most one of subtrees at these vertices, e.g., v_1, may be absent in $\mathbb{S}'$ (if $\mathcal{L}(\mathbb{S}_{v_1})$ and $\mathcal{L}(\mathbb{T})$ are disjoint). In that case by applying the same argument inductively on v_2, we reach a vertex that stays in $\mathbb{S}'$. Thus we have a vertex with similar cluster present in $\mathbb{S}'$. Therefore, $f_{\mathbb{S}'}(u) \neq 0$.

($\Leftarrow$) Since $f_{\mathbb{S}'}(u) \neq 0$, there exists a vertex v in $\mathbb{S}'$ such that $C_{\mathbb{T}}(u) = C_{\mathbb{S}'}(v)$. Now we must have $v \in I(\mathbb{S})$, since the restriction of $\mathbb{S}$ to $\mathcal{L}(\mathbb{T})$ does not introduce a new vertices in $\mathbb{S}'$. Thus, in $\mathbb{S}$, $\hat{C}_{\mathbb{T}}(v) = C_{\mathbb{T}}(u)$ (by the definition of $\mathbb{S}'$). Therefore, $f_{\mathbb{S}}(u) \neq 0$. □

Corollary 1. $RF(\mathbb{T}, \mathbb{S}) = |\mathcal{L}(\mathbb{T})| - |I(\mathbb{T})| + 2|\mathcal{F}_{\mathbb{S}}| - 2$.

Proof. In Lemma 2, $|I(\mathbb{S}')| = |\mathcal{L}(\mathbb{T})| - 2$. Now the result is trivially true. □

3.3.2 An LCA-Based Preprocessing Algorithm

We now describe a $O(n)$-time algorithm to compute the initial vertex function for a supertree $\mathbb{S}$ relative to input tree $\mathbb{T}$, along with the RF distance between these two trees.

Definition 3 (LCA Mapping). For $\mathbb{S}$ and $\mathbb{T}$, the *LCA mapping* $\mathcal{M}_{\mathbb{S},\mathbb{T}} : V(\mathbb{S}) \to V(\mathbb{T})$ is defined as

$$\mathcal{M}_{\mathbb{S},\mathbb{T}}(v) := \begin{cases} \text{LCA}_{\mathbb{T}}(\hat{C}_{\mathbb{T}}(v)), & \text{if } \hat{C}_{\mathbb{T}}(v) \neq \phi \ ; \\ \text{null}, & \text{otherwise.} \end{cases}$$

Fig. 3.3 illustrates LCA mappings.

Lemma 4. *For all* $u \in I(\mathbb{T})$, $f(u) = |B|$, *where* $B := \{v \in I(\mathbb{S}) : \mathcal{M}_{\mathbb{S},\mathbb{T}}(v) = u$ *and* $|C_{\mathbb{T}}(u)| = |\hat{C}_{\mathbb{T}}(v)|\}$.

Proof. By the definition of $f(u)$, it suffices to show that $B = U$, where $U := \{v \in I(\mathbb{S}) : C_{\mathbb{T}}(u) = \hat{C}_{\mathbb{T}}(v)\}$. If $v \in U$, then, by the definition of $\mathcal{M}_{\mathbb{S},\mathbb{T}}(v)$, $v \in B$. If $v \in B$, then $\mathcal{M}_{\mathbb{S},\mathbb{T}}(v) = u$ and $|C_{\mathbb{T}}(u)| = |\hat{C}_{\mathbb{T}}(v)|$ imply that $C_{\mathbb{T}}(u) = \hat{C}_{\mathbb{T}}(v)$. □

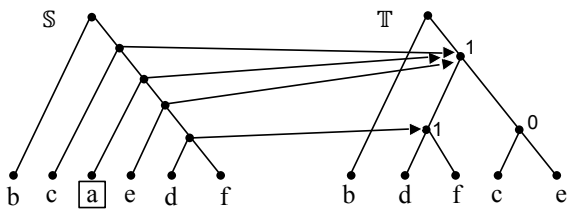

Figure 3.3 The LCA mapping from $\mathbb{S}$ to $\mathbb{T}$. Vertex a in $\mathbb{S}$ is mapped to *null* as $a \notin \mathcal{L}(\mathbb{T})$. The internal vertices of $\mathbb{T}$ are labeled with the values of the vertex function.

Lemma 4 proves that LCA mappings can be used to compute the vertex function for given trees $\mathbb{S}$ and $\mathbb{T}$. A vertex $u \in \mathbb{T}$ with $f(u) = 0$ indicates that the corresponding cluster does not match any restricted cluster in $\mathbb{S}$; such vertices of $\mathbb{T}$ compose the set $\mathcal{F}_{\mathbb{S}}$. Corollary 1 can now be directly applied to compute $RF(\mathbb{T}, \mathbb{S})$.

The LCA computation for $\mathbb{T}$ can be done in $O(n)$ time, and the LCA mapping from $\mathbb{S}$ to $\mathbb{T}$ can be done in $O(n)$ time (Bender and Farach-Colton, 2000). Further, from Lemmas 2–4 we can compute the RF distance between $\mathbb{S}$ and $\mathbb{T}$ in $O(n)$ time as well. We assume that there is a distinct rooted copy $\mathbb{S}$ of S for each input tree T, and that $\mathbb{S}$ and $\mathbb{T}$ are rooted according to the outgroup chosen for T.

3.4 Solving the NNI Search Problem

Let T be an arbitrary tree in $\mathcal{P}$. We now show how to compute the RF distance from T to each tree in the NNI neighborhood of a supertree S in linear time of the size of neighborhood. The key idea is to simulate each NNI operation on unrooted tree S on its rooted version $\mathbb{S}$, using the LCA mapping from $\mathbb{S}$ to $\mathbb{T}$ to quickly compute the RF distance. This mapping changes as NNI operations are performed on S, but we show that it can be updated in constant time at each step.

Consider an NNI move across an edge $e = \{x, y\}$ of S. Let A and B be the two subtrees on the x side of e, and C and D be the two subtrees on the y side of e (Fig. 3.4).

Observation 1. *The tree obtained from swapping A and C is isomorphic to the tree obtained from swapping B and D.*

Proof. Both swaps produce the same sets of splits and are therefore the trees are isomorphic

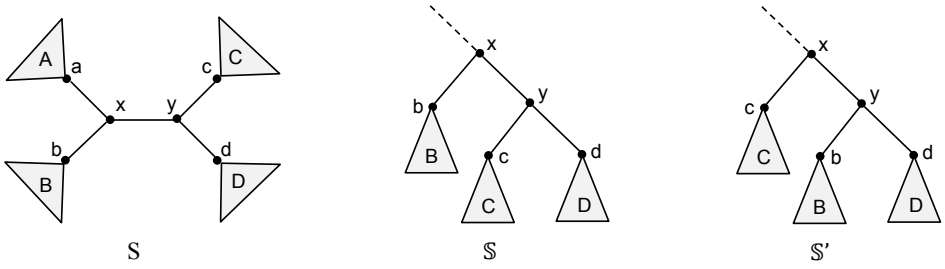

Figure 3.4 Unrooted tree S, the rooted version $\mathbb{S}$ and the result $\mathbb{S}'$ of an NNI operation swapping subtrees B and C. The figure assumes that the outgroup lies in subtree A.

by the Splits Equivalence Theorem. □

Without loss of generality, assume that the NNI move swaps B with C, resulting in tree S'. Also, assume that the subtrees A, B, C, and D connect with edge e through vertices a, b, c, and d, respectively. In $\mathbb{S}$, either x is the parent of y or y is the parent of x, depending on which side has the outgroup. In the first case, the children of y are c and d. Further, if the sibling of y is b, then the outgroup must be in subtree A (see Fig. 3.4), otherwise it is in subtree B. The other cases are analogous. Observe that the parent-child and sibling relationships can be checked in constant time.

Let the children of y in $\mathbb{S}$ be c and d, and the sibling of y be b. After the NNI operation the children of y are b and d, and the sibling of y is c. Let the resulting tree be called $\mathbb{S}'$. (Note that if outgroup was in B then we would have swapped A and D, since, from Observation 1 both operations produce the same result.)

Lemma 5. *(i) For all* $u \in I(\mathbb{S})\backslash\{y\}$, $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(u) = \mathcal{M}_{\mathbb{S},\mathbb{T}}(u)$ *and (ii)* $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(y) = LCA_{\mathbb{T}}(\mathcal{M}_{\mathbb{S},\mathbb{T}}(b), \mathcal{M}_{\mathbb{S},\mathbb{T}}(d))$.

Proof. (i) For $v \in V(\mathbb{S}'_b) \bigcup V(\mathbb{S}'_c) \bigcup V(\mathbb{S}'_d)$, $\mathbb{S}_v \simeq \mathbb{S}'_v$. Thus, $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(v) = \mathcal{M}_{\mathbb{S},\mathbb{T}}(v)$. Now, $\mathcal{L}(\mathbb{S}'_x) = \mathcal{L}(\mathbb{S}_x)$, thus $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(x) = \mathcal{M}_{\mathbb{S},\mathbb{T}}(x)$. Also, except for subtree $\mathbb{S}_x$, the rest of the tree remains the same in $\mathbb{S}'_x$, thus for $v \in V(\mathbb{S}')\backslash V(\mathbb{S}'_x)$, $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(v) = \mathcal{M}_{\mathbb{S},\mathbb{T}}(v)$.
(ii) Observe that, b, d are children of y in $\mathbb{S}'$, and $\mathbb{S}'_b \simeq \mathbb{S}_b$, $\mathbb{S}'_d \simeq \mathbb{S}'_d$. So, $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(y) = \mathrm{LCA}_{\mathbb{T}}(\mathcal{M}_{\mathbb{S}',\mathbb{T}}(b), \mathcal{M}_{\mathbb{S}',\mathbb{T}}(d)) = \mathrm{LCA}_{\mathbb{T}}(\mathcal{M}_{\mathbb{S},\mathbb{T}}(b), \mathcal{M}_{\mathbb{S},\mathbb{T}}(d))$. □

Let $h := \mathcal{M}_{\mathbb{S},\mathbb{T}}(y)$ and $h' := \mathcal{M}_{\mathbb{S}',\mathbb{T}}(y)$. Note that h and h' may refer to the same vertex in $\mathbb{T}$. Let G denote the set $\{w \in \{h, h'\} : f_{\mathbb{S}}(w) = 0, \text{ but } f_{\mathbb{S}'}(w) \geq 1\}$, and L the set $\{w \in \{h, h'\} :$

$f_{\mathbb{S}}(w) \geq 1,$ but $f_{\mathbb{S}'}(w) = 0\}$.

Lemma 6. *$RF(\mathbb{S}', \mathbb{T}) = RF(\mathbb{S}, \mathbb{T}) - 2|G| + 2|L|$.*

Proof. $RF(\mathbb{S}', \mathbb{T}) = |\mathcal{L}(\mathbb{T})| - |I(\mathbb{T})| - 2 + 2|\mathcal{F}_{\mathbb{S}'}| = |\mathcal{L}(\mathbb{T})| - |I(\mathbb{T})| - 2 \quad + 2|\{u \in I(\mathbb{T}) : f_{\mathbb{S}'}(u) = 0\}|$ $= |\mathcal{L}(\mathbb{T})| - |I(\mathbb{T})| - 2 + 2|\mathcal{F}_{\mathbb{S}}| - 2|\{u \in \{h, h'\} : f_{\mathbb{S}}(u) = 0 \ \& \ f_{\mathbb{S}'}(u) \geq 1\}| + 2|\{u \in \{h, h'\} : f_{\mathbb{S}'}(u) = 0 \ \& \ f_{\mathbb{S}}(u) \geq 1\}| = RF(\mathbb{S}, \mathbb{T}) - 2|G| + 2|L|$ □

Lemma 7. *The RF distance from T to any $S' \in NNI_S$ can be computed in $O(1)$ time.*

Proof. From Lemma 5, the LCA mapping of only one vertex y changes in $\mathbb{S}'$ and can be computed in constant time using the LCA pre-computation of $\mathbb{T}$. Also, the values of $f_{\mathbb{S}'}(h)$ and $f_{\mathbb{S}'}(h')$ can be updated in constant time. Finally, $RF(\mathbb{S}', \mathbb{T})$ is computed in constant time as shown in Lemma 12. Further, $RF(S', T) = RF(\mathbb{S}', \mathbb{T})$. □

Theorem 1. *The NNI Search problem can be solved in $\Theta(nk)$ time.*

Proof. There are $\Theta(n)$ edges in S. From Lemma 7, updating the RF distance after an NNI move takes constant time per input tree. Thus for k input trees it takes $\Theta(nk)$ time. Further, the pre-processing of Section 4 takes $\Theta(nk)$ time. □

3.5 Solving the 2-ECR Search Problem

As seen in Section 2, a 2-ECR operation on a binary tree consists of contracting two edges e_1 and e_2, and then refining the contracted tree into a binary tree. These two edges may or may not be adjacent edges in the tree. Our algorithm for 2-ECR Search handles each case separately.

3.5.1 Case 1: The edges are not adjacent

In this case we use the next result.

Lemma 8. *(Ganapathy et al. (2003)) Let T be an unrooted leaf-labeled tree and let T' be a 2-ECR neighbor of T such that the 2-ECR move involves the contraction and refinement of two non-adjacent edges in T. Then T' can be reached from T through two NNI moves.*

Thus, when e_1, e_2 are not adjacent, the optimal 2-ECR neighbor can be obtained by computing an optimal NNI neighbor of an NNI neighbor of S. There are $\Theta(n)$ NNI neighbors of S and, by Theorem 1, an optimal NNI neighbor can be found in $\Theta(nk)$ time. Therefore, we have the following result.

Lemma 9. *The optimal 2-ECR neighbor of an n-taxa supertree S for a profile $\mathcal{P}$ of k trees, subject to the restriction that the edges involved are not adjacent, can be computed in $\Theta(n^2k)$ time.*

3.5.2 Case 2: The edges are adjacent

Note that there are $O(n)$ possible pairs of adjacent edges for a tree with n leaves. For a given pair (e_1, e_2) of edges, the 2-ECR operation contracts e_1 and e_2 and creates a degree-5 vertex. It then refines this vertex in one of the 15 possible ways to obtain a new binary tree. We show that for each possible refinement the RF distance from an input tree can be computed in constant time.

Let $e_1 = \{x, y\}$ and $e_2 = \{y, z\}$ be the two edges in S chosen for the 2-ECR move. Let S' be the tree that results from the move. Let A and B be the subtree on x side of e_1, C be the subtree connected to y, and D and E be the subtree on z side of S. As in tree T_1 of Fig. **??**. Also, assume that the subtrees A, B, C, D, and E connect with e_1 and e_2 through vertices a, b, c, d, and e, respectively.

Note that in tree S, any of the five subtrees A, B, C, D, E can contain the outgroup.

1. *A has the outgroup:* Then x is the parent of y and y parent of z. The sibling of y is b.
2. *B has the outgroup:* Then x is the parent of y and y parent of z. The sibling of y is a.
3. *D has the outgroup:* Then z is the parent of y and y parent of x. The sibling of y is e.
4. *E has the outgroup:* Then z is the parent of y and y parent of x. The sibling of y is d.
5. *C has the outgroup:* Then y is the parent of x and z.

It is easy to check in constant time which case holds.

Now we divide the 15 possible S's into two categories for computing the RF distance from all possible S'.

3.5.2.1 Category 1: Subtree C does not change position

If C is fixed at the same place as S in S' then the remaining four subtrees can be arranged in three ways. Observe that one of them is identical to S so we do not consider it. In the other two cases, we swap a subtree on x side (A or B) with a subtree on z side (D or E). Notice that this move is similar to one NNI where the edge spans two edges e_1 and e_2. We show how to compute the RF distance of T from tree S', obtained by swapping A with D in S. The other case can be analyzed similarly.

First, we check which subtree among A, B, C, D, E contains the outgroup in $\mathbb{S}$. If A or D contains the outgroup, then we swap B with E. The splits obtained from swapping A and D are the same as the splits obtained from swapping B and E; thus, the trees are isomorphic.

Next, we find the vertices of $\mathbb{S}'$ with any change in LCA mapping in $\mathcal{M}_{\mathbb{S}',\mathbb{T}}$. Based on the topology of $\mathbb{S}$, there are three cases:

1. *x is the parent of y and y is the parent of z.* For all $t \in I(\mathbb{S}')\backslash\{y, z\}$, $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(t) = \mathcal{M}_{\mathbb{S},\mathbb{T}}(t)$. Further, $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(z) := LCA(\mathcal{M}_{\mathbb{S},\mathbb{T}}(a), \mathcal{M}_{\mathbb{S},\mathbb{T}}(e))$, and $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(y) := LCA(\mathcal{M}_{\mathbb{S},\mathbb{T}}(c), \mathcal{M}_{\mathbb{S}',\mathbb{T}}(z))$.

2. *y is the parent of x and z.* For all $t \in I(\mathbb{S}')\backslash\{x, z\}$, $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(t) = \mathcal{M}_{\mathbb{S},\mathbb{T}}(t)$. Further, $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(z) := LCA(\mathcal{M}_{\mathbb{S},\mathbb{T}}(a), \mathcal{M}_{\mathbb{S},\mathbb{T}}(e)$, and $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(x) := LCA(\mathcal{M}_{\mathbb{S},\mathbb{T}}(d), \mathcal{M}_{\mathbb{S},\mathbb{T}}(b))$.

3. *z is the parent of y and y is the parent of x.* For all $t \in I(\mathbb{S}')\backslash\{y, x\}$, $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(t) = \mathcal{M}_{\mathbb{S},\mathbb{T}}(t)$. Moreover, $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(x) := LCA(\mathcal{M}_{\mathbb{S},\mathbb{T}}(d), \mathcal{M}_{\mathbb{S},\mathbb{T}}(b))$, and $\mathcal{M}_{\mathbb{S}',\mathbb{T}}(y) := LCA(\mathcal{M}_{\mathbb{S},\mathbb{T}}(c), \mathcal{M}_{\mathbb{S}',\mathbb{T}}(x))$.

It can be checked in constant time which one of the above three cases holds; thus, the LCA mappings can be updated in constant time, too. Let H be the set defined as

$$H := \{u \in I(\mathbb{T}) : f_{\mathbb{S}'}(u) \neq f_{\mathbb{S}}(u)\}.$$

Observe that set H has at most four vertices and that it can be computed in constant time. The new RF score is computed from the change in the f values of the vertices in H in the

following way. For $t \in H$, if $f_{\mathbb{S}}(t) \geq 1$ and $f_{\mathbb{S}'}(t) = 0$ then the RF distance increases by 2 for t. Conversely, if $f_{\mathbb{S}}(t) = 0$ and $f_{\mathbb{S}'}(t) \geq 1$ then the RF distance decreases by 2 for t. Thus we have shown how the RF distance between a input tree and S', in Category 1, can be computed in constant time.

3.5.2.2 Category 2: Subtree C changes position

In this case the place of C in S' can be occupied by A, B, D, or E. Further, in each case the remaining four subtrees can be arranged at vertices x and z in three ways. Thus there are 12 possibilities in this Category. We generate all S's in this in an order that helps us to compute RF distance easily. First, we perform one NNI that swaps subtree C with a subtree from {A, B, D, E} and compute the RF distance for the generated S'. For this S', we swap one subtree from x side with one subtree from z side to generate the other two S's. We describe how to do so for subtree A; the same can be done for the rest of the subtrees.

Once again, our algorithm first checks the topology of $\mathbb{S}$. If A or C has the outgroup, we swap the subtrees other than A and C from x and y side of e_1. Observe that this is an NNI operation, and so the RF distance between T and S' can be computed in constant time from Lemma 7. The next two moves on S' are similar to Category 1. Thus, for each tree the RF distance can be computed in constant time.

Summarizing the analyses for Categories 1 and 2, we have the following.

Lemma 10. *The optimal 2-ECR neighbor of an n-taxa supertree S for a profile $\mathcal{P}$ of k trees, subject to the restriction that the edges involved are adjacent, can be computed in $\Theta(nk)$ time.*

Lemmas 9 and 10 give us the next result.

Theorem 2. *The 2-ECR Search problem can be solved in $\Theta(n^2k)$ time.*

3.6 p-ECR

As seen in Section 2, a p-ECR operation on a binary tree consists of contracting p edges, and then refining the contracted tree into a binary tree. For the case when p is a variable we have the next result.

Theorem 3. *The p-ECR Search problem for arbitrary p cannot be solved in polynomial time unless P = NP.*

Proof. Follows from the observation that the Unrooted RF Supertree problem reduces to the p-ECR Search problem by letting $p = n - 3$. □

The techniques used for the 1-ECR (NNI) and 2-ECR Search problems can generalize naturally to the 3-ECR Search problem, and further p-ECR for any fixed p. We limit ourselves to outlining of the main ideas of 3-ECR Search problem, which gives some insight into the p-ECR Search problem and its limitations as p grows. We rely on the following fact (see, e.g., (Semple and Steel, 2003) for a proof).

Lemma 11. *The number of unrooted binary phylogenetic trees on n leaves is* $(2n-5)!! = (2n-5)\cdot(2n-7)\cdots 5\cdot 3\cdot 1$.

There are three subcases to consider, depending on the adjacency relationships among the edges chosen for the 3-ECR operation:

1. **No two of the three edges are adjacent.** There are $O(n^3)$ sets of three such edges in S. In this case, the 3-ECR operation first produces three degree-four vertices, each of which is then refined independently of the others in three possible ways. Thus, there are 27 possible full refinements of the contracted graph. The problem now becomes one of finding an optimal NNI neighbor of an NNI neighbor of an NNI neighbor of S. By Theorem 1, this can be done in $O(n^3)$ time.

2. **Only two of the three edges are adjacent.** There are $O(n)$ pairs of adjacent edges and $O(n)$ single edges in S, for a total of $O(n^2)$ possibilities. As seen in the previous section, the node resulting from contracting the two adjacent edges can be refined in 15 possible ways. The node resulting from contracting the third edge can be refined in three possible ways, for a total of 45 possibilities for each such triple of edges. In this case, the best 3-ECR neighbor can be computed in $O(n^2)$ time by enumerating the $O(n)$ 1-ECR neighbors, and among them, finding the best 2-ECR neighbor for the case where the two edges involved are adjacent. By Lemma 10, the total time for doing this is $O(n^2k)$.

3. **The three edges form a subtree.** A contraction of three edges generates one degree-6 node, which, by Lemma 11, can be refined in $(2 \cdot 6 - 5)!! = 105$ ways. There are $O(n)$ sets of three edges which form a subtree in S. Through a detailed case analysis, one can show that the RF distance of the supertree to each input tree can be obtained in constant time, after a linear amount of preprocessing. Thus in this case the optimal 3-ECR neighbor can be computed in $O(nk)$ time; however, the O-notation hides a large constant.

To summarize, the total time needed for the 3-ECR Search problem is $O(n^3)$, but the constant factors are significantly higher than they are for 2-ECR.

3.7 Experimental Results

We implemented our unrooted RF heuristic based on 2-ECR local search and tested it on simulated and empirical data sets. (The executable code is available by request from the first author.) In our analyses, we first ran the SPR-based rooted RF supertree local search program (RRF) (Bansal et al., 2010b) on each data set. Next, we selected the best supertree from the output as the starting supertree for our unrooted RF supertree program (URF). We then compared the results of URF search with RRF and MRP. MRP was carried out on an Intel Core 2 Duo 2.4 GHz Macintosh laptop with 4GB of main memory; for rest of the analyses, we used an Intel Pentium Core 2 Duo 2.6 GHz desktop computer with 4 GB of main memory.

Local search algorithms can get caught in local optima with relatively high RF-distance scores. To alleviate this problem, we implemented a search heuristic based on the parsimony ratchet (Nixon, 1999). Our ratchet search performs 18 iterations, where each iteration consist of two local 2-ECR searches. The first search takes as its input the current supertree and a randomly selected subset of 10% of the input trees. The supertree produced by the first search is then used as the starting point for the second search, which works with the original input trees. The output of the iteration is the supertree produced by the second search.

We used simulated data sets produced by SMIDGEN (Swenson et al., 2010, 2011), available at www.cs.utexas.edu/~phylo/datasets/supertrees.html. In brief, they are 100-taxon, 500-taxon, and 1000-taxon datasets, and comprised of *mixed* source trees, containing one "scaf-

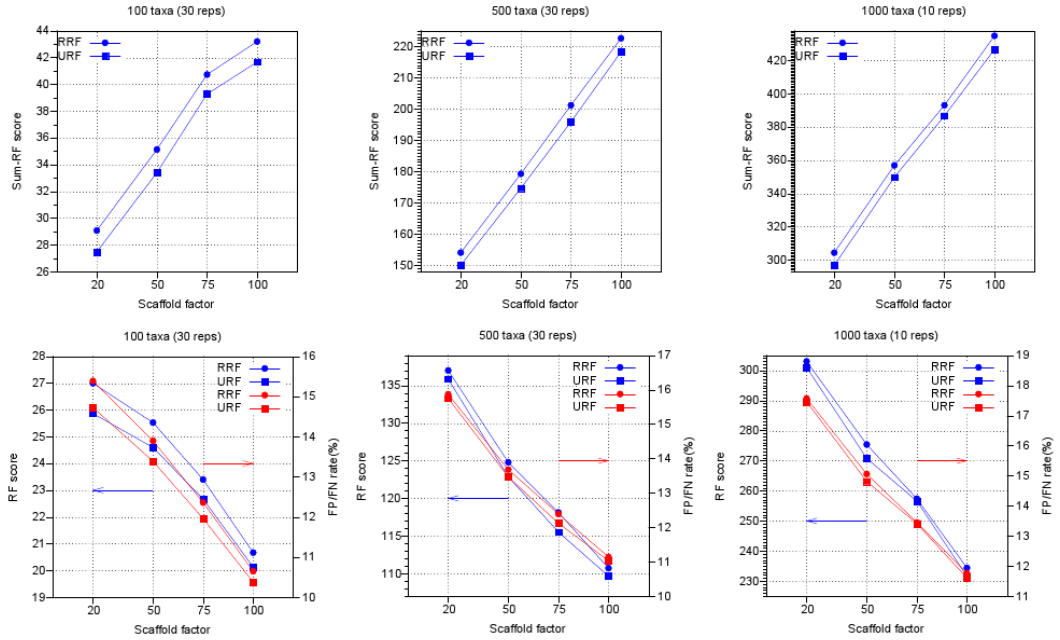

Figure 3.5 Graphs on mixed source tree datasets with 100, 500, and 1000 taxa. The top three graphs show the Sum-RF scores for RRF and URF as a function of the scaffold density. The bottom three graphs display the RF score using the y1-axis, and FN/FP rate using the y2-axis, both as a function of the scaffold density. Each data point represents the average of the scores from all 30 model conditions (10 for the 1000-taxon data set).

fold" dataset and several clade-based datasets. The clade-based datasets are produced by dense taxon sampling within a rooted subtree. A scaffold dataset is constructed by randomly selecting the taxa from the entire dataset with a probability called the "scaffold factor", ranging from 20% to 100%. The total number of taxa and the scaffold factor determines the model condition of a supertree study. For each model condition the source tree file has 6, 16, and 26 trees for 100, 500, and 1000 taxa data sets, respectively. There are 30 replicates for each model condition, except for the 1000-taxon data sets that have 10 replicates. See (Swenson et al., 2010, 2011) for full detail of these datasets.

Topological error for each estimated supertree was evaluated according to RF score, false positive/false negative rate, and Sum-RF score. Let T be an estimated supertree, T_0 be the true (model) tree, and Γ be the profile of source trees. Then,

- the *RF score* is $RF(T, T_0)$,

Table 3.1 Running Time for Simulated Datasets

Num. Taxa	Scaffold Factor (%)	RRF-Time	URF-Time
100	20	4s	7s
	50	5s	8s
	75	5s	8s
	100	5s	8s
500	20	10m 40s	20m 2s
	50	11m 29s	17m 17s
	75	15m 36s	13m 38s
	100	10m 7s	23m 8s
1000	20	1h 53m 21s	2h 57m 12s
	50	2h 25m 37s	3h 30m 1s
	75	2h 36m 28s	4h 35m 36s
	100	2h 53m 6s	5h 24m 31s

- the *false positive (FP) rate* is $\frac{|\Sigma(T_0)\backslash\Sigma(T)|}{|\Sigma(T)|}$ and the *false negative (FN) rate* is $\frac{|\Sigma(T)\backslash\Sigma(T_0)|}{|\Sigma(T_0)|}$,
- the *Sum-RF score* is $RF(\Gamma, T)$.

Observe that when the estimated tree and true tree are binary, as is the case for the simulated data, then the false positive rate and false negative rate are equal. Thus, for simulated data we refer to the "FP/FN" rather than to the two rates separately. Note also that, for binary trees, the (normalized) RF score is twice the FP/FN rate.

We show the RF scores and FP/FN rates of RRF and URF for different scaffold values and taxa size in Fig. 3.5. Figure 3.5 also gives the Sum-RF of RRF and URF for different scaffold values and taxa size. The running times are summarized in Table 6.1. We did not run MRP on the simulated data, as MRP and RRF are already compared in (Swenson et al., 2011).

We also evaluated the performance of our implementation on five empirical data sets, and compared it with RRF and MRP. The data included published supertree data sets for marsupials (Cardillo et al., 2004), placental mammals (Beck et al., 2006), and dinosaurs (Lloyd et al., 2008). Additionally, we used data sets we assembled from the gymnosperm and Saxifragales plant clades. To construct these data sets, first all core nucleotide sequence data from these

clades, with some outgroups, were downloaded from GenBank (www.ncbi.nlm.nih.gov). The sequences were clustered into sets of homologs based on BLAST scores (Altschul et al., 1990), and the clusters were aligned using MUSCLE Edgar (2004). The gene trees were inferred with maximum likelihood using RAxML (Stamatakis, 2006a). The gymnosperm data set has 77 gene trees that include sequences from a total of 950 species, and the Saxifragales data set has 51 gene trees and 958 total species. Although these data sets are unpublished, similar gene tree data sets were used in a previous study (Wehe and Burleigh, 2010). The input gene trees are available on the Dryad data repository (www.dryaddata.org). Our experimental results are summarized in Table 6.2.

Because RRF requires that the source trees be rooted, we rooted the unrooted trees at the midpoint of the longest leaf-to-leaf path before passing them to RRF. The dinosaur, gymnosperm, and Saxifragales data sets, as well as all the simulated data sets, had unrooted input trees.

For our MRP analyses, we used the parsimony ratchet implemented in PAUP* (Swofford, 2003). First we build the MRP matrix using r8s (Sanderson, 2003). In our search, the starting trees were generated from a random sequence addition followed by one round of TBR swapping. Once a local optimum was reached, we performed 20 ratchet iterations. Each ratchet iteration randomly reweighted 10% of the characters with weight 1.0, while keeping the weight of other characters at 0. A round of TBR hill-climbing was performed on the reweighted data matrix. During the second round, the weight of all characters was returned to 1.0, and another round of TBR hill-climbing was performed. This returns a set of trees, each of which has the best (found) score. Next, we compute the greedy consensus (gMRP) tree for this set. The greedy consensus is a refinement of the majority consensus, and thus, it contains all the bipartitions present in more than half the input trees.

We note that in our experiments, FN/FP rates of the resulting RRF supertrees are noticeably better than those reported in (Swenson et al., 2011), where the tests are done on the same simulated datasets (see Fig. 3.5). Additionally, the URF supertree heuristic produces supertrees of even lower FN/FP rates, and thus outperforms all the supertree techniques studied in (Swenson et al., 2011), at least by this measure.

Table 3.2 Experimental Results for Empirical Datasets

Data Set	Supertree Method	RF Distance	Time
Dinosaurs	RRF	12188	6h 24m 50s
(420 taxa;	URF	**11500**	14h 47m 31s
165 trees)	MRP	11512	8m 54s
Gymnosperms	RRF	4370	2d 11h 50m 11s
(950 taxa;	URF	**4054**	2d 52m 37s
78 trees)	MRP	4420	35m 41s
Marsupials	RRF	1353	1h 4m
(272 taxa;	URF	**1333**	1h 39m 10s
156 trees)	MRP	1335	1m 42s
Placental Mammals	RRF	5431	24m 37s
(116 taxa;	URF	**5391**	1h 42m 42s
726 trees)	MRP	5393	1m 3s
Saxifragales	RRF	2156	12h 29m 27s
(959 taxa;	URF	**1992**	15h 55m 33s
51 trees)	MRP	2196	47m 5s

Also, in our experiments on simulated data, we observed that trees with small RF distance from the source trees also had, on the average, lower RF distance to the true tree (see Fig. 3.5). This is clearly a desirable characteristic for any distance measure used in supertree construction.

For all five empirical data sets, URF notably reduced the RF score of the starting supertree and always performed better than MRP (Table 6.2). In contrast to that, in simulated tests our URF supertree heuristic marginally improved on RRF in some cases. One reason for this may be that URF performs well when the dataset has a large number of unrooted trees and consequently more instances of incorrect rooting. In such situations, the rooted technique generates supertrees that are more distant from the optimal supertrees. In contrast, URF does not rely on any rooting and thus it performs well compared to RRF.

3.8 Conclusion

The RF supertree problem directly seeks a supertree that is most similar to input trees based on the RF distance, making it a desirable and potentially useful approach for building comprehensive phylogenies. Until now, the only existing heuristics for RF supertrees required rooted input Bansal et al. (2010b). However, nearly all recent supertree studies have included unrooted input trees (e.g., Beck et al. (2006); Bininda-Emonds et al. (2007); Cardillo et al. (2004)).

Thus, our new heuristics for the unrooted RF supertree problem greatly extend the utility of the RF supertree method. Further, our experiments demonstrate that they can easily handle data sets with nearly 1000 taxa, while improving upon the quality of rooted RF supertrees. The improvement could be especially significant when the number of input trees is large, and incorrect rooting of one or more input trees is a serious possibility. This suggests that the RF supertree method is a viable alternative to MRP for a wide range of data sets.

Still, there are several directions for future development. In our experiments, the unrooted heuristic started from a high quality supertree (the rooted RF supertree). Although this strategy appears to be effective, it is also costly. Further tests are needed to examine the effects of the starting tree on the performance of the unrooted heuristic and to identify fast and effective strategies to build a starting tree. It is also important, and appears to be relatively straightforward, to incorporate uncertainty within the input trees into an RF supertree analysis by weighting the splits when calculating the RF distance.

CHAPTER 4. Inferring Species Trees from Incongruent Multi-Copy Gene Trees Using the Robinson-Foulds Distance

4.1 Introduction

Constructing species phylogenies from a collection of gene trees requires summarizing and reconciling the phylogenetic information contained in the genomic data. To achieve this, the majority of existing species tree reconstruction methods typically use a model of gene evolution that reconciles the gene tree and species tree topologies based on a specific evolutionary process, such as duplication and loss or deep coalescence. These models doubtlessly simplify the true processes of genome evolution. A gap remains, though, as these models do not reflect the complexity of the evolutionary processes of many genes, that are affected by multiple evolutionary processes. Adapting more complex and realistic models can quickly become unwieldy, making it hard or impossible to analyze large genomic data sets, potentially prohibited scientists from obtaining the good estimates of the evolutionary relationships from the available genomic data sets. Here we consider the problem of constructing species tree from gene trees as a problem of identifying the dominant phylogenetic signals among the incongruent gene trees, without attempting to hypothesize the processes that caused the incongruence among gene trees. Such simplified process may avoid undesirable evolutionary assumptions while allowing the user to include large gene tree data sets in a phylogenetic analysis.

Existing methods for inferring species trees from collections of gene trees can be divided into two broad categories: non-parametric methods based on gene tree parsimony (GTP), and likelihood-based approaches (Ané et al., 2007; Kubatko et al., 2009; Liu and Pearl, 2007). GTP methods take a collection of discordant gene trees and try to find the species tree that implies the fewest evolutionary events. GeneTree (Page, 1998), DupTree (Wehe et al., 2008), and

DupLoss (Bansal et al., 2010a) seek to minimize the number of duplications or duplications and losses. GeneTree (Page, 1998), Mesquite (Maddison, 1997), PhyloNet (Yu et al., 2011), and the method of (Bansal et al., 2010a) minimize deep coalescence events. The Subtree Prune and Regraft (SPR) supertree method (Whidden et al., 2012) is based on minimizing the number of LGT events, and thus it can be considered a type of gene tree parsimony. Some of these methods have fast and effective heuristics, enabling the analysis of very large data sets. However, errors in the gene trees can mislead GTP analyses (Burleigh et al., 2011; Huang and Knowles, 2009; Sanderson and McMahon, 2007a). Also, in some cases GTP methods may be statistically inconsistent even when the gene tree topologies are correct (Than and Rosenberg, 2011). Most of the likelihood-based methods use coalescence models to reconcile gene tree topologies (Kubatko et al., 2009; Liu and Pearl, 2007). Although such likelihood-based approaches have a strong statistical foundation, they can be computationally expensive or intractable.

While all the existing methods differ widely in their details, at a high level, except (Ané et al., 2007), they all are based on potentially restrictive assumptions about the source of discordance among gene trees. For example, GTP based on a duplication and loss cost assumes that all differences between a gene tree and the species tree are caused by either gene duplications or losses. However, gene tree estimation error plays a significant role in the conflict among gene trees (e.g., Rasmussen and Kellis (2011)). Further, these errors in gene trees can drastically increase the estimated number of duplications and losses (Burleigh et al., 2009; Hahn, 2007; Rasmussen and Kellis, 2011) and lead to error in the species tree inference (Burleigh et al., 2009; Sanderson and McMahon, 2007b).

We present a tree distance metric based approach for constructing species tree from discordant multi-copy gene trees. The main advantage of using a tree distance metric in species tree construction is that the resulting method is not restricted to any specific evolutionary process of gene tree discordance. This allows the species tree to be estimated even when the conflict among gene trees is the result of many evolutionary processes and gene tree estimation errors. Our specific distance measure is a generalization of the Robinson-Foulds (RF) distance measure to multi-labeled trees (mul-trees), i.e., trees where multiple leaves can have the same

label. The ability to use mul-trees as input, instead of being restricted to single copy genes, allows this method to incorporate the wealth of genomic data from multi-copy genes, not only single-copy genes, into phylogenetic inference. Our new MulRF method takes as input a collection of multi-copy gene trees (or mul-trees) and finds a species tree at minimum RF distance to the input gene trees (Section 4.2). The RF distance has been used in the supertree method for singly-labeled input trees (Bansal et al., 2010b; Chaudhary et al., 2012b), and the tree distance based maximum-likelihood supertree approach has been proven to be statistically consistent (Steel and Rodrigo, 2008). Moreover, our method has the scalability and accuracy expected for genome-wide analyses of many species.

MulRF is a NP-hard problem. Therefore, a heuristic algorithm is required to estimate solutions for large data sets. We provide a fast $\Theta(n^2k)$-time algorithm for the MulRF problem, where n is the total number of distinct leaves in the input collection of gene trees and k is the number of gene trees (Section 4.3). We implemented the MulRF heuristic and examined the performance of this method on gene tree simulations that incorporate gene tree error, gene duplication and loss, and/or lateral gene transfer (Section 4.4).

4.2 MulRF Problem

A *profile* is a tuple of multi-copy gene trees $\mathcal{P} := (\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_k)$, also called *input mul-trees*, where $\mathcal{T}_i = (T_i, M_i, \varphi_i)$ for each $i \in \{1, \ldots, k\}$. A *species tree* or a *tree* for $\mathcal{P}$ is a singly-labeled phylogenetic tree S such that $\mathcal{L}(S) = \bigcup_{i=1}^{k} M_i$. We write n to denote $|\mathcal{L}(S)|$, the total number of distinct leaves in the profile. In this Chapter, we assume that the size of each input mul-tree differs only by a constant factor from the size of the resulting species tree.

We extend the notion of RF distance to the case where $\mathcal{L}(T_1) \subseteq \mathcal{L}(T_2)$ by letting $RF(T_1, T_2) := RF(T_1, T_{2|\mathcal{L}(T_1)})$. We define the *RF distance* from a profile $\mathcal{P}$ to a species tree S for $\mathcal{P}$ as $RF(\mathcal{P}, S) := \sum_{\mathcal{T} \in \mathcal{P}} RF(\mathcal{T}, S)$.

Let $\mathcal{B}(\mathcal{P})$ be the set of all binary species trees for $\mathcal{P}$.

Problem 3 (RF for MUL-Trees (MulRF)).

Input: A profile $\mathcal{P} = (\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_k)$ of unrooted, mul-trees.

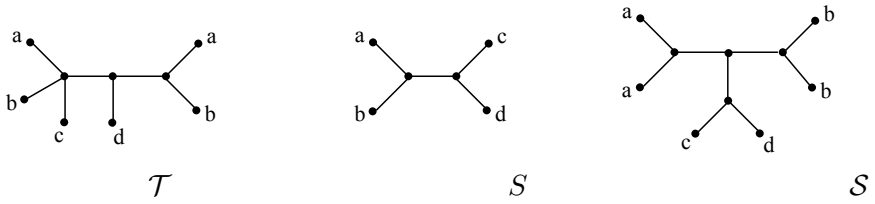

Figure 4.1 Input mul-tree $\mathcal{T}$ and the species tree S. The extended species tree $\mathcal{S}$ is also shown.

Output: A species tree S^* for $\mathcal{P}$ such that $RF(\mathcal{P}, S^*) = \min_{S \in \mathcal{B}(\mathcal{P})} RF(\mathcal{P}, S)$.

The MulRF problem is NP-hard even when all the input mul-trees are singly-labeled trees on the same leaf set (McMorris and Steel, 1993). In fact, just computing the RF distance between two mul-trees is hard (Chapter 7 Section 7.1). Nevertheless, we now show that it is straightforward to compute the RF distance between an input mul-tree and a species tree.

Let $\mathcal{T} = (T, M, \varphi)$ be an input mul-tree and S be a species tree, where $M \subseteq \mathcal{L}(S)$. The *extended species tree* is the mul-tree $\mathcal{S}$ constructed from S by replacing each $a \in \mathcal{L}(S)$ by an internal node connecting to k leaves labeled with a, where $k := |\varphi^{-1}(a)| > 1$. See Fig. 4.1.

A *full differentiation* of $\mathcal{T}$ is a leaf labeled tree $\mathbf{T}$ such that T and $\mathbf{T}$ are isomorphic.

Let $\mathcal{T} = (T, M, \varphi)$ and $\mathcal{S} = (T', M', \varphi')$ be two unrooted mul-trees. Two full differentiations $\mathbf{T}$ and $\mathbf{S}$ of $\mathcal{T}$ and $\mathcal{S}$, respectively, are *consistent* if for each $a \in M \cap M'$, $\tau_1(\varphi^{-1}(a)) = \tau_2(\varphi'^{-1}(a))$, where T and $\mathbf{T}$ are isomorphic under bijection $\tau_1 : V(T) \to V(\mathbf{T})$ and T' and $\mathbf{S}$ are isomorphic under bijection $\tau_2 : V(T') \to V(\mathbf{S})$. For instance, a consistent full differentiation can be obtained by relabeling each of the k copies of each leaf label a by $a_1, a_2, \ldots, a_k$ in both the mul-trees.

Theorem 4 (Ganapathy et al. (2006))**.** *Let $\mathcal{T}$ and $\mathcal{S}$ be two mul-trees. Then, $RF(\mathcal{T}, \mathcal{S}) = \min\{RF(\mathbf{T}, \mathbf{S}) : \mathbf{T}$ and $\mathbf{S}$ are mutually consistent full differentiations of $\mathcal{T}$ and $\mathcal{S}$, respectively$\}$.*

Theorem 5. *Let $\mathcal{T}$ be an input mul-tree and $\mathcal{S}$ be the extended species tree. Then, all mutually consistent full differentiations of $\mathcal{T}$ and $\mathcal{S}$ give the same RF distance.*

Proof. Let the given input mul-tree $\mathcal{T}$ is such that $\mathcal{T} := (T, M, \varphi)$. We prove the Theorem by showing that for each $a \in M$, where $|\varphi^{-1}(a)| = k$, all $k!$ ways of uniquely relabeling corresponding k leaves in both $\mathcal{T}$ and $\mathcal{S}$ result into the same number of matched and unmatched

splits in the corresponding mutually consistent full differentiations. The set of splits in $\mathcal{T}$ can be divided into two categories:

- *Category 1:* Splits that have all the leaves labeled with a in one part. Such a split will always have a match irrespective of the labeling.
- *Category 2:* The remaining splits. Such splits are not present in $\mathcal{S}$, therefore, they will never have a match irrespective of the labeling. □

In short, the RF distance between an input mul-tree and a species tree can be computed by 1) extending the species tree, 2) producing one consistent full differentiation of the two mul-trees, and 3) applying the split based formula to compute the RF distance.

4.3 Solving the MulRF Problem

Our local search heuristic for the MulRF problem starts with an initial species tree and explores the space of possible species trees in search of a *locally optimum* species tree; i.e., a tree whose score is minimum within its "neighborhood". The neighborhood is defined in terms of the *SPR* operation (Allen and Steel, 2001). The set of all trees obtained by the application of a single SPR operation on T is called the *SPR neighborhood* of T, and is denoted by SPR_T. The size of this neighborhood is $\Theta(n^2)$.

Problem 4 (SPR Search).
Input: A profile $\mathcal{P} = (\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_k)$ of unrooted, mul-trees and a binary, species tree S for $\mathcal{P}$.
Output: A tree $S^* \in SPR_S$ such that $RF(\mathcal{P}, S^*) = \min_{S' \in SPR_S} RF(\mathcal{P}, S')$.

In Section 4.3.1, we present an algorithm for the SPR search problem that runs in time $\Theta(n^2k)$. The algorithm relies on results from Chapter 3 Section 3.3, which characterize the RF distance between unrooted trees in terms of least common ancestors in rooted versions of those trees. These properties enable us to update the RF distance quickly after an SPR operation has been applied to one of the trees.

4.3.1 Solving the SPR Search Problem

Let $\mathcal{T} = (T, M, \varphi)$ be an arbitrary mul-tree in $\mathcal{P}$. We now show how to compute the RF distance from $\mathcal{T}$ to each tree in the SPR_S neighborhood in linear time of the size of the neighborhood. Let $\mathcal{S}$ be the extended species tree S after extending for $\mathcal{T}$. Let $\mathbf{T}$ and $\mathbf{S}$ be any two mutually consistent full differentiations of $\mathcal{T}$ and $\mathcal{S}$, respectively. By Theorem 5, computing the RF distance between an input mul-tree $\mathcal{T}$ and all trees in the SPR neighborhood of an extended supertree $\mathcal{S}$ reduces to finding the RF distance between $\mathbf{T}$ and each tree in the SPR neighborhood of $\mathbf{S}$.

Suppose an SPR operation on S cuts the edge $e = \{x, y\}$, and that X, Y are the subtrees of $S - e$ containing x, y, respectively. Suppose subtree Y is pruned and regrafted by the same cut edge to a new vertex obtained by subdividing an edge in X. The degree-two vertex x is suppressed and the new vertex is denoted by x. Observe that there are $O(n)$ possible edges in X to regraft Y. We perform regrafts in an order that leads to a constant time RF distance computation for each successive regraft.

Observation 1. *For $Z \in \{X, Y\}$, if $M \cap \mathcal{L}(Z) = \emptyset$, then $RF(S', \mathcal{T}) = RF(S, \mathcal{T})$ for each S' obtained from S by regrafting Y on any edge in X.*

Proof. Let the extension of S be $\mathcal{S} := (T', M', \varphi')$. Let $\mathbf{S}$ be a full differentiation of $\mathcal{S}$ that is consistent with $\mathbf{T}$, where T' and $\mathbf{S}$ are isomorphic under bijection $\tau : V(T') \to V(\mathbf{S})$.

For $Z \in \{X, Y\}$, let $\mathbf{S}[Z] = \{l \in \mathcal{L}(\mathbf{S}) : \varphi'(\tau^{-1}(l)) \in \mathcal{L}(Z)\}$.

Since, $\mathcal{L}(\mathbf{S}_{|\mathcal{L}(\mathbf{T})}) \cap \mathbf{S}[Z] = \emptyset$, $RF(\mathbf{S}_{|\mathcal{L}(\mathbf{T})}, \mathbf{T}) = RF(\mathbf{S}'_{|\mathcal{L}(\mathbf{T})}, \mathbf{T})$. Now, $RF(\mathcal{S}, \mathcal{T}) = RF(\mathbf{S}, \mathbf{T}) = RF(\mathbf{S}_{|\mathcal{L}(\mathbf{T})}, \mathbf{T}) = RF(\mathbf{S}'_{|\mathcal{L}(\mathbf{T})}, \mathbf{T}) = RF(\mathbf{S}', \mathbf{T}) = RF(\mathcal{S}', \mathcal{T})$. □

We begin by regrafting Y at an edge incident to a leaf in X. Let $\overline{S}$ and $\overline{\mathbf{S}}$ denote, respectively, the tree that results from performing the prune-and-regraft and the full differentiation of this result tree. We compute the RF distance between $\mathbf{T}$ and $\overline{\mathbf{S}}$ using the algorithm described in the previous section. This method works by computing the RF distance between the rooted trees $\mathbb{T}$ and $\overline{\mathbb{S}}$ obtained by rooting $\mathbf{T}$ and $\overline{\mathbf{S}}$ at any leaf labeled by an element of $M \cap \mathcal{L}(X)$.

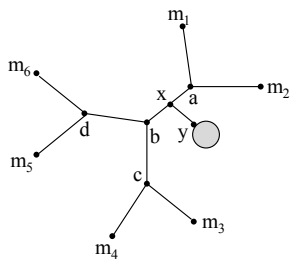

Figure 4.2 A tree with a subtree regrafted at edge $\{a, b\}$. One iteration of vertices in the tree is $m_1, a, m_2,\ a, b, c, m_3, c, m_4, c, b, d, m_5,\ d, m_6, d, b, a, m_1$. The resulting ordering $\aleph$ is $\{m_1, a\}$, $\{a, m_2\}$, ..., $\{a, m_1\}$.

(Note that, by Observation 1, if $M \cap \mathcal{L}(X) = \emptyset$, then $\mathcal{T}$'s distance from $\overline{S}$ is same as S.) The algorithm also computes the LCAs for $\mathbb{T}$ and the LCA mapping from $\overline{\mathbb{S}}$ to $\mathbb{T}$.

We perform the remaining regrafts of Y on edges in X by iterating through the vertices of X, starting from a leaf and exploring as far as possible along each branch before backtracking. The k^{th} regraft is performed on the edge between the k^{th} and $k + 1^{st}$ vertices in this iteration. Let us denote this ordering of edges by $\aleph$. See Fig. 4.2. Observe that each two distinct consecutive edges in $\aleph$ are adjacent. We will show that, after the initial RF distance computation for $\overline{S}$, we can compute in constant time the RF distance for the result of regrafting on each successive (adjacent) edges in $\aleph$.

Beginning with $\overline{S}$, each $S' \in SPR_S$ helps in computing the RF distance of the next tree in the above regraft order. Assume that $S' \in SPR_S$ results from regrafting Y at edge $\{a, b\}$ in X as shown in Fig. 4.2. Let the rooted tree obtained after extending and differentiating S' be denoted by $\mathbb{S}'$. The LCA mapping and RF distance have been computed for $\mathbb{S}'$. Let $S'' \in SPR_S$ denote the tree obtained by regrafting Y on edge $\{b, c\}$ in X and the rooted counterpart of S'' is $\mathbb{S}''$.

Next, we find the vertices of $\mathbb{S}''$ whose LCA mapping $\mathcal{M}_{\mathbb{S}'',\mathbb{T}}$ has changed as a result of the SPR operation. Based on the topology of $\mathbb{S}'$, there are three cases:

1. *x is parent of b and b is parent of c.* For all $t \in I(\mathbb{S}'')\backslash\{x, b\}$, $\mathcal{M}_{\mathbb{S}'',\mathbb{T}}(t) = \mathcal{M}_{\mathbb{S}',\mathbb{T}}(t)$. Further, $\mathcal{M}_{\mathbb{S}'',\mathbb{T}}(b) := \mathcal{M}_{\mathbb{S}',\mathbb{T}}(x)$, and $\mathcal{M}_{\mathbb{S}'',\mathbb{T}}(x) := \text{LCA}(\mathcal{M}_{\mathbb{S}',\mathbb{T}}(c), \mathcal{M}_{\mathbb{S}',\mathbb{T}}(y))$.

2. *b is parent of c and x.* For all $t \in I(\mathbb{S}'')\backslash\{x\}$, $\mathcal{M}_{\mathbb{S}'',\mathbb{T}}(t) = \mathcal{M}_{\mathbb{S}',\mathbb{T}}(t)$. Further, $\mathcal{M}_{\mathbb{S}'',\mathbb{T}}(x) := \text{LCA}(\mathcal{M}_{\mathbb{S}',\mathbb{T}}(c), \mathcal{M}_{\mathbb{S}',\mathbb{T}}(y))$.

3. *b is parent of x and c is parent of b.* For all $t \in I(\mathbb{S}'')\backslash\{b, x\}$, $\mathcal{M}_{\mathbb{S}'',\mathbb{T}}(t) = \mathcal{M}_{\mathbb{S}',\mathbb{T}}(t)$. Moreover, $\mathcal{M}_{\mathbb{S}'',\mathbb{T}}(x) := \mathcal{M}_{\mathbb{S}',\mathbb{T}}(b)$, and $\mathcal{M}_{\mathbb{S}'',\mathbb{T}}(b) := \text{LCA}(\mathcal{M}_{\mathbb{S}',\mathbb{T}}(d), \mathcal{M}_{\mathbb{S}',\mathbb{T}}(a))$.

Since we can check in constant time which one of the above three cases holds, the LCA mappings can be updated in constant time too. Let H be a set $\{u \in I(\mathbb{T}) : f_{\mathbb{S}''}(u) \neq f_{\mathbb{S}'}(u)\}$. Set H can be computed in constant time. Observe that H has at most four vertices. Let G denotes the set $\{w \in H : f_{\mathbb{S}'}(w) = 0, \text{ but } f_{\mathbb{S}''}(w) \geq 1\}$, and L denote the set $\{w \in H : f_{\mathbb{S}'}(w) \geq 1, \text{ but } f_{\mathbb{S}''}(w) = 0\}$.

Lemma 12. $RF(\mathbb{S}'', \mathbb{T}) = RF(\mathbb{S}', \mathbb{T}) - 2|G| + 2|L|$.

Proof.

$$\begin{aligned} RF(\mathbb{S}'', \mathbb{T}) &= |\mathcal{L}(\mathbb{T})| - |I(\mathbb{T})| - 2 + 2|\mathcal{F}_{\mathbb{S}''}| \\ &= |\mathcal{L}(\mathbb{T})| - |I(\mathbb{T})| - 2 \\ &\quad + 2|\{u \in I(\mathbb{T}) : f_{\mathbb{S}''}(u) = 0\}| \\ &= |\mathcal{L}(\mathbb{T})| - |I(\mathbb{T})| - 2 + 2|\mathcal{F}_{\mathbb{S}'}| \\ &\quad - 2|\{u \in H : f_{\mathbb{S}'}(u) = 0 \ \& \ f_{\mathbb{S}''}(u) \geq 1\}| \\ &\quad + 2|\{u \in H : f_{\mathbb{S}''}(u) = 0 \ \& \ f_{\mathbb{S}'}(u) \geq 1\}| \\ &= RF(\mathbb{S}', \mathbb{T}) - 2|G| + 2|L| \end{aligned}$$ □

Thus, after the initial regraft of Y at a leaf in X, we can compute in constant time the RF-distance between $\mathbf{T}$ and the species tree that results from each subsequent regraft.

Lemma 13. *For each $\{x, y\} \in E(S)$, where X and Y are two resulting subtrees containing x and y, respectively. The RF distance for the set of trees obtained by regrafting X (resp. Y) on each edge in Y (resp. X) can be computed in $\Theta(n)$ time.*

Proof. The RF distance computation for $\overline{S}$, obtained by pruning Y and regrafting at a leaf in X, can be done in $\Theta(n)$ time. After $\overline{S}$, the RF distance for each tree S', obtained by regrafting Y on each edge in X, can be computed in constant time by performing regrafts in the order of $\aleph$. There are $\Theta(n)$ edges in $\aleph$, thus the RF computation for all the trees can be done in $\Theta(n)$ time. The same argument applies for pruning X and regrafting on the edges in Y. □

Theorem 6. *The SPR Search problem can be solved in* $\Theta(n^2k)$ *time.*

Proof. There are $\Theta(n)$ internal edges in S. For each edge $\{x, y\}$ in S, where X, Y be two resulting subtrees containing x, y, respectively. The RF distance for all the trees obtained by regrafting X (or Y) on each edge in Y (or X) can be computed in $\Theta(n)$ time from Lemma 13. Thus for k input trees the RF distance can be checked in $\Theta(nk)$ time. The total time over all $\Theta(n)$ internal edges is $\Theta(n^2k)$. □

4.4 Experimental Evaluation

4.4.1 Method

Simulated data sets. We generated model species trees using the uniform speciation (Yule) module in the program Mesquite (Maddison and Maddison, 2009). Two sets of model trees were generated: i) 50 taxa trees of height 220 thousand years (tyrs), ii) 100 taxa trees of height 440 tyrs (note that the dates are relative; they do not have to represent thousands of years). Each data set had 20 model species trees. We evolved 150 and 300 gene trees for each 50- and 100-taxon model species tree, respectively. We used duplication-loss model by Arvestad et al. (2003) to evolve gene trees within the model tree. We applied LGT events on the evolved gene trees, using the standard subtree transfer model of LGT. One LGT event causes the subtree rooted at a vertex c to be pruned and regrafted at an edge (a, b), where a and b together are not in the path from the root (of the tree) to c. We used gene duplication and loss (D/L) rate of 0.002 events/gene per tyrs and LGT rate of 2 events per gene tree. In other words, a gene tree can have 0 to 2 LGT events.

We evolved gene trees based on four evolutionary scenarios: i) no duplications, losses, or LGT (called *none*), ii) D/L rate 0.002 and no LGT (called *dl*), iii) no duplication or loss, and LGT rate 2 (called *lgt*), and iv) D/L rate 0.002 and LGT rate 2 (called *both*). The parameter values for each simulation are called the *model condition*. We deleted 0 to 25% of the taxa (selected at random) from each gene tree to represent missing data, which is common in almost all phylogenomic studies. For each gene tree, we used Seq-Gen (Rambaut and Grassly, 1997) to simulate a DNA sequence alignment of length 500 based on the GTR+Gamma+I model.

Num. Taxa	Sets	Only-dup	Dup-loss	SPRS	MulRF
50	*none*	< 1s	2s	8h 34m 32s	3s
	lgt	< 1s	2s	8h 30m 30s	2s
	dl	< 1s	3s	NA	6s
	both	< 1s	3s	NA	6s
100	*none*	9s	37s	21h 34m 25s	58s
	lgt	11s	49s	19h 6m 9s	51s
	dl	9s	30s	NA	1m 11s
	both	11s	37s	NA	1m 15s

Table 4.1 Running time for species tree estimations

The parameters of the model were chosen with equal probability from the parameter sets estimated in (Ganapathy, 2006) on three biological data sets (Swenson et al., 2010). We estimated maximum likelihood trees from each simulated sequence alignment using RAxML (Stamatakis, 2006a), performing searches from 5 different starting trees and saving the best tree. We rooted each estimated gene tree at the midpoint of the longest leaf-to-leaf path before the species tree construction.

Species tree estimation. We estimated species trees via GTP minimizing only the number of duplications (Only-dup) (Wehe et al., 2008), GTP minimizing duplications and losses (Dup-loss) (Bansal et al., 2010a), GTP minimizing LGT events (SPR supertree or SPRS for short) (Whidden et al., 2012), and the MulRF heuristic. Both Only-dup and Dup-loss were executed with their default settings, including a fast leaf-adding heuristic for initial species tree construction. SPRS was run with 25 iterations of the global rearrangement search option. For 50-taxon data sets, it calculated the exact rSPR distance if it was 15 or less, and otherwise it estimated the rSPR distance using the 3-approximation. For the 100-taxon data sets, we used the 3-approximation of the rSPR distance. SPRS does not allow mul-trees as input. Therefore we only ran it on *none* and *lgt* data sets. Experiments were performed on the University of Florida High Performance Computing test nodes with 8 to 24 cores.

Performance evaluation. We report the average topological error (ATE) for each model condition. This is the average of the normalized RF distance (dividing the RF distance by number of internal edges in both trees) between each of the 20 model species trees and their

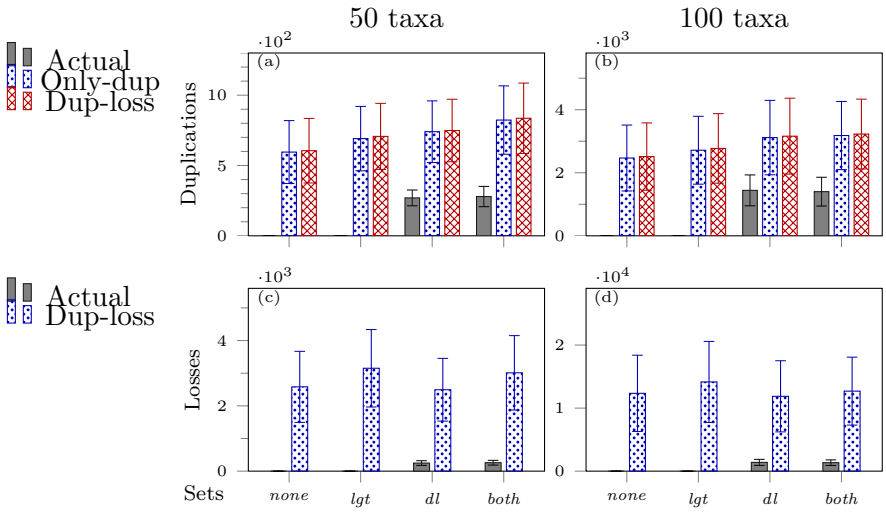

Figure 4.3 Graphs a-b shows duplications estimated by Only-dup and Dup-loss, and Graphs c-d losses estimated by Dup-loss, against the actual number of these events in gene trees, for all model conditions; means and standard errors are shown.

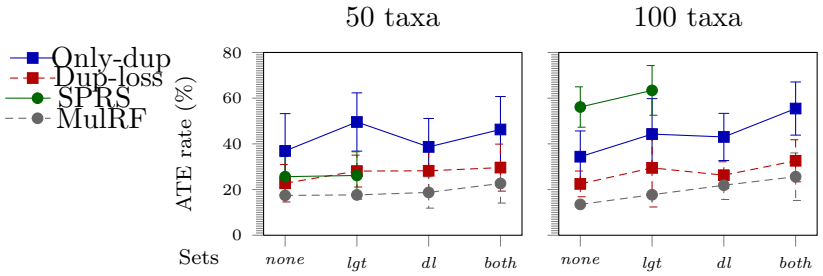

Figure 4.4 Average topological error (means with standard error bars) for species tree constructed by Only-dup, Dup-loss, SPRS, and MulRF method, for all model conditions.

estimated species trees. An ATE of 0 indicates that two trees are identical, and an ATE of 100 indicates that two trees share no common splits. We also compared the number of gene duplications estimated by Only-dup and Dup-loss and losses estimated by Dup-loss with the actual number of these events in each gene tree simulation.

4.4.2 Results

Both Dup-loss and Only-dup overestimate duplications for sets *dl* and *both* in both 50- and 100-taxon model trees (Fig. 4.3(a,b)). They also imply many duplications in the *none* and *lgt* data sets, where the simulations included no duplications. Similarly, Dup-loss overestimates losses for sets *dl* and *both* and also erroneously estimates losses for sets *none* and *lgt* (Fig. 4.3(c,d)).

For each set of 50- and 100-taxon model trees, the MulRF species trees are more accurate than those produced by the other three methods. For example, the ATE rate of MulRF is 16.75% to 39.91% lower than the method of lowest ATE rate among other three methods (Fig. 4.4).

In order to examine how Only-dup, Dup-loss, and SPRS methods perform when the process of gene tree evolution only includes events that these methods assume to be the source of discordance, we simulated gene trees that using a model that includes only duplication and loss, or LGT. While SPRS could not be tested on the former, Only-dup and Dup-loss had high ATE rate (indicating low accuracy) on the latter.

The MulRF software as well as simulated data are freely available for download at http://genome.cs.iastate.edu/CBL/MulRF/.

4.5 Conclusion

We presented the new MulRF method for inferring species tree from incongruent gene trees that is based on RF distance metric. Unlike most previous phylogenetic methods using gene trees, our approach is based on a generic tree distance metric, freeing it from potentially restrictive assumptions about the causes of the conflict among gene trees. As a result it is appealing for analyses of genomic data sets, in which many processes such as deep coalescence, recombination, gene duplications and losses, and LGT, as well as phylogenetic error likely contribute to gene tree discord.

Simulation experiments allowed us to evaluate the accuracy of our method by comparing it against the true species tree, something that cannot be done on real data. We compared the species trees constructed by MulRF and GTP methods that consider only duplication (Wehe et al., 2008), duplication and loss (Bansal et al., 2010a), and only LGT (Whidden et al., 2012) with the true species trees. Likelihood-based methods were not considered because the simulated gene trees were comparatively large in size for these methods. In all experiments, MulRF produced trees that are more similar to the true species trees than those obtained by other three methods. Further, our algorithm ran quickly on moderate-size data sets, finishing in under two minutes on data sets containing 300 gene trees evolved over 100 taxon species

trees, suggesting it is scalable for large-scale phylogenomic analyses.

One reason of MulRF method's strong performance may be its "unrooted" tree distance metric (rather than "rooted" as in GTP methods). Gene tree reconstruction methods, for example maximum likelihood or gene tree parsimony, typically generate unrooted trees. Rooting the gene trees is in itself a difficult problem in phylogenetics (e.g., Smith (1994); Wheeler (1990); Sanderson and Shaffer (2002)). GTP methods may be affected by erroneous rootings, while this would not affect a method based on unrooted tree distance metric like MulRF. However, the rooted metric of a GTP method gives a benefit of inferring a rooted species tree, which is a limitation of the unrooted MulRF method.

The simulation experiments also provided us the opportunity to study how accurately the GTP methods hypothesize the evolutionary history of the simulated genes. The simplest measure compares the number of GTP estimated duplications and losses with the actual number of these events for each simulated gene tree. Whether the conflict among the simulated gene trees was solely due to gene tree error or gene tree error in the presence of duplication and loss, and/or LGT events, the estimated duplications and losses always differed greatly from the actual values. These results provide us with a novel insight: if the models based on evolutionary processes are inaccurate, then there is reason to explore phylogenetic methods based on tree distances that are not based on specific evolutionary processes. The MulRF method shows the potential of such an approach.

There are several directions for future development. First and foremost, more tests are needed to characterize the performance of MulRF methods under different evolutionary scenarios. Another future direction will be to incorporate estimates of gene tree uncertainty into the supertree analysis by weighing the splits differently when computing the RF distance. The effectiveness of the MulRF method in inferring species trees from multi-copy gene trees suggests that other tree distance metrics in the same context. A natural candidate for study is the quartet distance. Future work should also evaluate the suitability of different distance metrics in estimating species trees under different error models and evolutionary scenarios.

CHAPTER 5. A Simulation Study to Compare Two Non-parametric Approaches for Species Trees Construction

5.1 Introduction

Inferring species relationships in the presence of discordant gene histories is a major challenge for modern phylogenetics. An effective method for such phylogenomic analyses must address the variety of causes of gene tree incongruence while remaining computationally tractable for large genomic data sets. Among the existing methods, although numerous recent studies have used GTP to infer phylogenies from genomic data (Sanderson and McMahon (2007b); Holton and Pisani (2010); Burleigh et al. (2011); Medina et al. (2011); Ness et al. (2011); Katz et al. (2012); Near et al. (2012); Wainwright et al. (2012)), there have been few formal studies to evaluate the performance of GTP, especially using genes with a history of duplication and loss. One concern is that GTP methods usually address a single biological process (e.g., gene duplication/loss, deep coalescence, or LGT) and implicitly assume that all incongruence among gene trees is caused by the specified process. In fact, much conflict among gene trees likely results from error in the gene tree inference (e.g., Rasmussen and Kellis (2011)), which can drastically inflate estimates of the number of duplications and losses (Hahn (2007); Burleigh et al. (2009); Rasmussen and Kellis (2011)) and mislead GTP (Sanderson and McMahon (2007b); Burleigh et al. (2009)). Finally, even given accurate gene trees that have evolved only under a single process, GTP in some cases may be inconsistent; that is, the GTP solution may converge to the incorrect species tree with the addition of more data (Than and Rosenberg, 2011).

A growing number of probabilistic (maximum likelihood or Bayesian) approaches based on coalescence models have been developed to infer species trees from potentially conflicting genes (e.g., Liu and Pearl (2007); Liu (2008); Kubatko et al. (2009); Heled and Drummond

(2010); see Liu et al. (2009)). Although promising, these approaches still are designed to address only orthologous sequences coalescence processes. Alternatively, Bayesian concordance analysis (Ané et al., 2007) estimates a species tree without making assumptions about the reason for gene discordance; however, it also is not currently designed to handle multi-copy gene trees. Probabilistic models of gene duplication and loss (Arvestad et al. (2004); Arvestad et al. (2009); Åkerborg et al. (2009); Górecki et al. (2011); Rasmussen and Kellis (2011)), or duplication, loss, and coalescence (Rasmussen and Kellis (2012)), have been developed to map gene duplications and losses on a fixed species tree, and some of these models simultaneously infer the gene tree topology (Arvestad et al. (2004); Arvestad et al. (2009); Åkerborg et al. (2009); Rasmussen and Kellis (2011)). However, simply calculating the maximum likelihood of these models with a single gene and a fixed species tree can be extremely computationally expensive, and such models have not been incorporated into phylogenetic inference.

MulRF, introduced in Chapter 4, is a non-parametric approach for combining multi-copy gene trees to infer a species tree. Like GTP methods, the input for the MulRF method is a collection of gene trees. Using a version of the RF distance generalized to multi-copy gene trees, the MulRF method seeks a species tree with the smallest RF distance to the collection of gene trees (Chapter 4). Thus, like GTP based on duplications and losses, the MulRF method can include genes with paralogs; however, in contrast to GTP, the MulRF approach does not attempt to reconcile the gene trees based on a specific biological process. Rather, it provides a mechanism-free approach for reconciling gene trees by simply seeking the species tree that is most similar to the input gene trees based on the generalized RF distance. Intuitively, the MulRF approach may be more appropriate than GTP if much of the conflict among genes is due to error or multiple, interacting biological processes, and preliminary simulation experiments suggest that a MulRF heuristic can with high accuracy resolve species trees from large sets of genes with a history of duplication and loss and limited LGT.

In this study, we address the question of which phylogenetic method can best resolve a species tree from multi-copy genes. We use gene simulations to evaluate the performance of GTP under duplication and duplication and loss cost models and compare them to the mechanism-free MulRF method. We look at the effects of species and gene sampling, gene tree

error, and missing, or unsampled, sequences, on the accuracy of the phylogenetic inference. Our results highlight the difficulty of inferring species trees from multi-copy genes, especially when there are high rates of duplication and loss, and raise concerns about the performance of GTP methods, especially GTP based only on minimizing duplications. They also demonstrate that in many cases, a method based on a generic tree distance measure, like the MulRF method, may provide more accurate estimates of the phylogeny than GTP.

5.2 Methods

5.2.1 Simulation

We conducted a series of simulation experiments to evaluate the performance of the MulRF heuristic and to compare its performance with GTP based on minimizing the gene duplication (Only-Dup) or duplication and loss (Dup-loss) cost. In brief, we first generated species trees using a Yule (pure birth) process simulation. Next, we generated gene trees inside model species trees of 50-, 100-, 250-, and 500-taxa using a model of gene duplication and loss. For each gene tree, we simulated an alignment of DNA sequences, and we estimated the maximum likelihood (ML) gene tree from this alignment. We performed the MulRF and GTP analyses using as input either the actual gene trees or the estimates of the gene tree topologies from the ML analysis. The performance of each of the three methods was evaluated based on the similarity of the estimated species trees to the original model species tree. For the GTP methods, we also evaluated the accuracy of the duplication and loss estimates.

5.2.1.1 Generating Model Species Trees

We generated model species trees using the "Uniform Speciation" (Yule, or pure birth, process) module in Mesquite (Maddison and Maddison, 2009). This creates a tree with a specified number of terminal taxa and a fixed time between the root and the present. We generated species trees with 50, 100, 250, and 500 taxa, and corresponding heights 220, 440, 800, and 1200 million years, respectively (note that the times are relative and the time units are arbitrary). For each 50- and 100-taxon model tree, we generated 40 species trees, and for

each 250- and 500-taxon model tree, we generated 20 species trees.

5.2.1.2 Simulating Gene Trees

We simulated 400 gene trees for each model species tree. Gene sequences often are either intensively sampled from clades of closely related species (e.g., primates) or from only a few, distantly related taxa which are selected to represent major lineages throughout a large clade. Following (Swenson et al., 2010), we refer to the first strategy as *clade-based* sampling and the second as *scaffold* sampling. For each model species tree, we generated 4 scaffold gene trees, and 396 clade-based gene trees. While the genes for inferring scaffold trees span the root of the model species tree, the genes for clade-based trees have a single birth node within the model species tree, which was randomly selected using the model tree topology and branch lengths.

To simulate the gene tree inside the model species tree, we used the duplication-loss model developed by Arvestad et al. (2003), which is based on the birth-death (BD) process (Feller, 1968). The BD process is a continuous-time process that generates a binary tree according to a constant rate of lineage bifurcation (gene duplication) and lineage termination (gene loss). We used gene duplication and loss (D/L) rates of 0.002, 0.004, and 0.008 events/gene per million years, following the D/L rates estimated from a primate data set (Rasmussen and Kellis, 2012). Each *model condition* is indicated by the number of taxa in the model species tree and the D/L rate used in simulating gene trees over it. Since gene sampling is rarely complete, we deleted 0 to 25% of the taxa (determined by randomly selecting a number between 0 and 25) from each gene tree, while ensuring each gene tree has at least 4 sequences from available taxa.

5.2.1.3 Simulating DNA Sequences and Building Input Trees

For each gene tree, a nucleotide sequence alignment of length 500 was simulated under the GTR+Gamma+I model using Seq-Gen (Rambaut and Grassly, 1997). The parameters of the model were chosen with equal probability from the parameter sets estimated by Ganapathy (2006) on three biological data sets (Swenson et al., 2010). Genes were simulated at fast, medium, or slow rates, implemented by rescaling the branch lengths of gene trees by a factor of 2.0, 1.0, or 0.1, respectively. While the genes for scaffold gene trees were always slow, 25% of

the genes of clade-based gene trees were slow, 50% medium, and 25% fast. For each sequence alignment, we estimated the maximum likelihood (ML) tree using RAxML (Stamatakis, 2006a), performing searches from 5 different starting trees and taking the best tree.

5.2.1.4 Species Tree Estimation

We conducted three types of analyses to evaluate the performances of Only-dup, Dup-loss, and MulRF methods. In the first analysis we ran Only-dup, Dup-loss, and MulRF on the gene trees that were simulated from the model trees. This analysis is performed before the nucleotide sequence simulation; therefore, we call it "pre-sequence" analysis. Note that in pre-sequence analysis, the gene trees have the correct root and their topologies have no error. The next two analyses were performed after simulating the nucleotide alignments and estimating ML gene trees. Also, they differ in the way they deal with the rooting of the gene trees. The Only-dup and Dup-loss methods require rooted gene trees, while MulRF does not. The ML analysis outputs an unrooted (or arbitrarily rooted) gene tree. In our first analysis, called "post-sequence (UR)", we feed the unrooted ML gene trees directly to all the three methods. Only-dup and Dup-loss run in their *unrooted settings* (Wehe et al. (2008); Burleigh et al. (2011)). In this setting, after a local SPR-search, they reroot the input trees (i.e., to change the artificial root of unrooted tree) to minimize the duplication or duplication and loss scores, and a new local-SPR search is performed. This procedure is repeated until re-rooting does not reduce the reconciliation cost. The second analysis, called "post-sequence (MR)", roots the ML gene trees using midpoint rooting implemented in Retree (Felsenstein, 1993) before passing them to the three methods. In the post-sequence (MR) and the pre-sequence analyses, Only-dup and Dup-loss use the rooted input trees and do not run in unrooted setting.

5.2.1.5 Performance Evaluation

We examined the accuracy of the three methods by comparing the estimated species tree with the original model species tree. We estimated the average topological error (ATE) percentage for each model condition by computing the average of the normalized RF distance between each model tree and the estimated species tree and multiplying by 100. The normalized RF

distance between a model tree and the estimated species tree is the RF distance divided by number of internal edges in both trees. An ATE of 0 indicates two trees are identical, and an ATE of 100 indicates the two trees share no common splits.

We also compared the number of gene duplications estimated by Only-dup, and duplications and losses estimated by Dup-loss with the actual number of these events in each gene tree simulation.

5.2.1.6 Additional Simulation Experiments

We performed two additional simulation experiments to examine the effects of gene tree sampling and sequence sampling within gene trees, respectively. In the first, we performed the species tree analyses using first 100 simulated gene trees for each model condition instead of 400 gene trees used in the original experiments. Therefore, we call it "100 gene trees experiment", in contrast to the original "400 gene trees experiment". This experiment allows us to examine the effect of the number of gene trees on species tree estimation.

In the second "incomplete sampling experiment" we simulated 200 gene trees with D/L rate 0.002 events/gene per million years over 100 taxon model species trees. From each gene tree, we deleted 0-25%, 25-50%, or 50-75% of the total sequences (randomly selecting a number from the specified range), while ensuring finally each model species tree has 200 simulated gene trees, that each contain at least 4 sequences from available taxa. For both these simulation experiments, pre-sequence and post-sequence (MR) analysis were performed.

5.3 Results

5.3.1 Accuracy of Species Tree Estimates

In both the 400 and 100 gene tree simulation experiments, the MulRF method always performs better than Only-dup or Dup-loss when the species trees are smaller ($\leq$ 100 taxa), and the Dup-loss method is generally more accurate than the other methods when the species trees are larger ($\geq$ 250 taxa; Figs. 5.1, 5.2). In all simulation experiments, the Only-dup method built trees that were less accurate than the Dup-loss or MulRF methods (e.g., Figs.

5.1-5.3).

The accuracy of the gene trees and the D/L rate affect the performance of all methods, but they have little effect on the relative accuracy of the different methods. In all simulation experiments, all methods are most accurate in the pre-sequence analyses, when the input gene tree topologies have no error (Figs 5.1-5.3). The impact of gene tree error is evident in the higher ATE values in the post-sequence analyses compare to their pre-sequence counterparts (Figs. 5.1-5.3). Among the post-sequence analyses, the two GTP methods perform best when the gene trees are rooted with mid-point rooting (post-sequence (MR)) compared to using unrooted gene trees and examining alternate rootings after each local SPR search (post-sequence (UR)). MulRF, which uses unrooted gene trees, performs similarly in the post-sequence (MR and UR) experiments. For all methods, no matter what gene trees are used, increased D/L rates also decreases accuracy, or increase ATE values (Figs. 5.1, 5.2).

The ATE rates of all three methods also increases as the sequence sampling decreases (Fig. 5.3(a)). This trend particularly affects the accuracy of species-tree estimates in the post-sequence (MR) analyses. For example, the Dup-loss and Only-dup supertrees share less than half of their splits with the model species tree in the 50-75% deletion case (Fig. 5.3(a)). However, increasing the number of input gene trees appears to improve species tree estimates. Across all 100 input tree analyses, in 93% of the species tree estimates with 100 genes, the ATE rates were higher than in the corresponding analysis using 400 gene trees (Figs. 5.1, 5.2). This effect was particularly pronounced for MulRF method, where the percent increase in MulRF's ATE rate was higher than the percent increase in the ATE rates of other two methods in 91.67% of the model conditions for post-sequence (MR) analyses (Figs. 5.1, 5.2).

5.3.2 Accuracy of Duplication and Loss Estimates

In the pre-sequence analyses for all experiments, Only-dup and Dup-loss always estimate fewer duplications than occurred in the simulations (Figs. 5.4, 5.5, 5.3(b)). Similarly, the Dup-loss method always underestimated the number of losses in the analyses using pre-sequence, or true, gene trees (Figs. 5.6, 5.7, 5.3(c)). Furthermore, the percent underestimation of duplications and losses in the pre-sequence analyses increases with D/L rate (Figs. 5.4-5.7). For

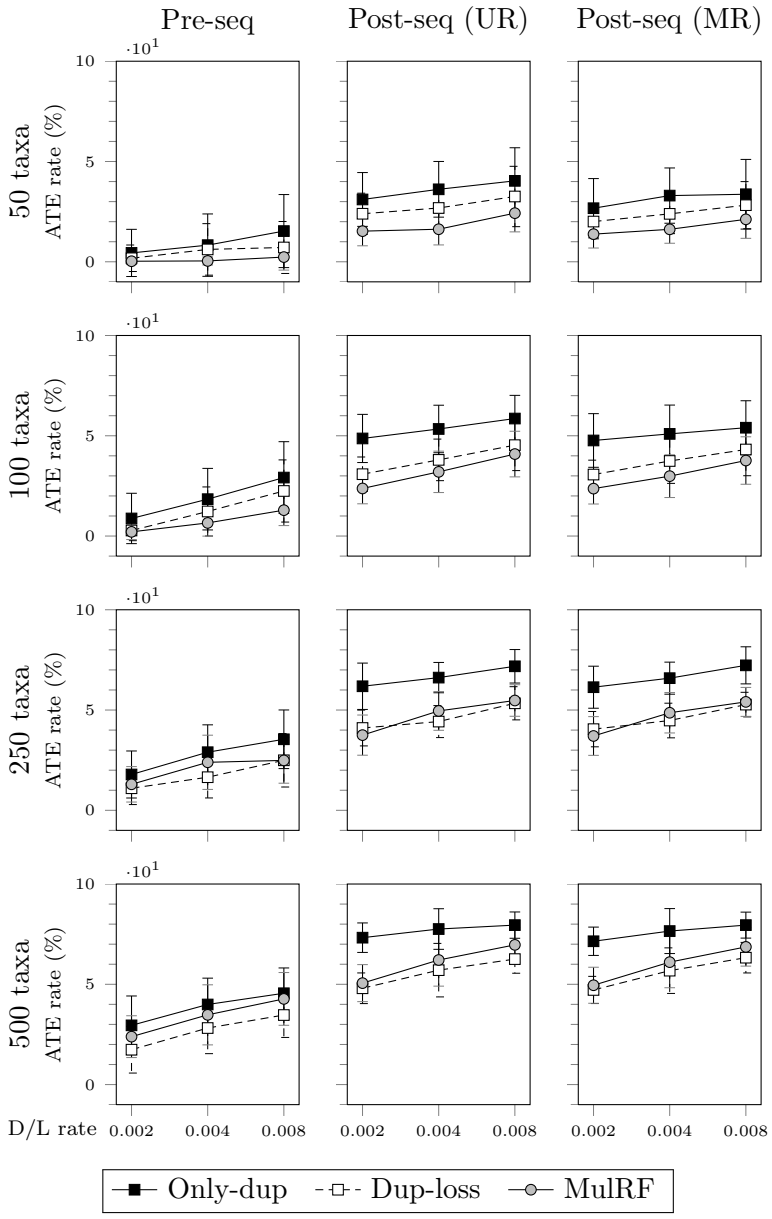

Figure 5.1 Results from the 400 gene trees experiments showing ATE rates of species trees constructed by Only-dup, Dup-loss, and MulRF methods in the pre-sequence and post-sequence (UR and MR) analyses, for differing D/L rates across 50, 100, 250, and 500 taxa model trees. Mean and standard errors are shown. Lower ATE rates mean higher accuracy.

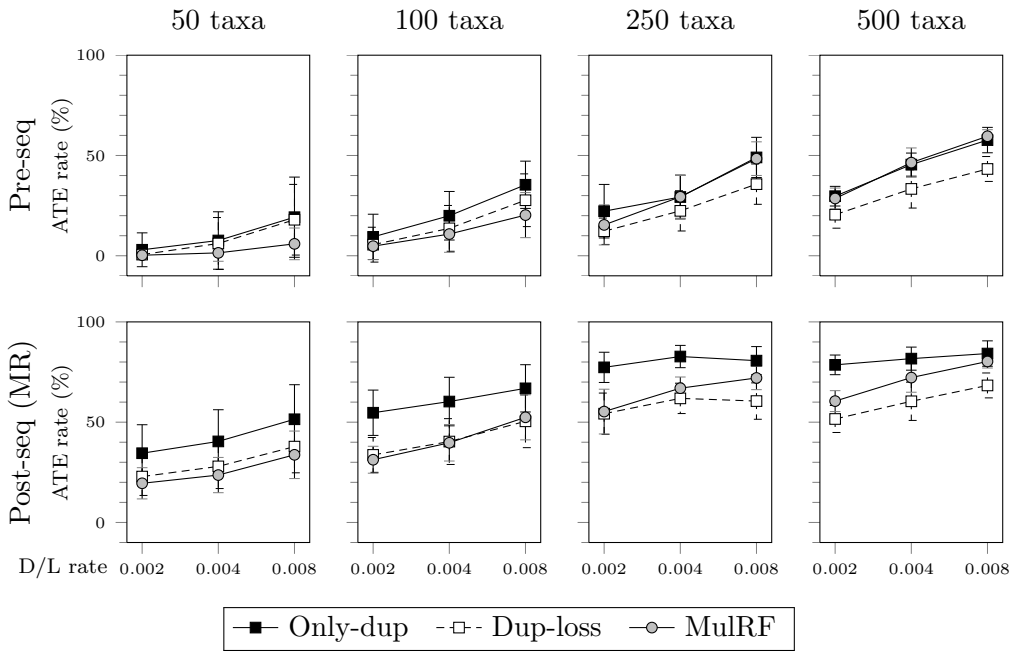

Figure 5.2 Results from the 100 gene trees experiments comparing the ATE rates for species trees constructed by Only-dup, Dup-loss, and MulRF methods in the pre-sequence and post-sequence (MR) analyses for differing D/L rates across 50, 100, 250, and 500 taxa model trees. Mean and standard errors are shown.

example, in the 50-taxon data set, Only-dup underestimates the number of duplications by 24.62% when the D/L rate 0.002 and by 44.18% when the D/L rate 0.008 (Fig. 5.4).

In contrast, in the post-sequence analyses, both Only-dup and Dup-loss often overestimate duplications (Figs. 5.4, 5.5) and Dup-loss often overestimates losses (Figs. 5.6, 5.7). However, the D/L rate in the simulations affects the accuracy of duplication and loss estimates. Estimates of duplications were relatively higher with low D/L rates than they were with high D/L rates (Figs. 5.4, 5.5). For example, with a D/L rate of 0.02, duplications are always overestimated, but with a higher D/L rate, they were sometimes underestimated (Figs. 5.4, 5.5). Similarly, the estimates of losses were relatively high with a low D/L rate and decrease with the increasing D/L rate (Figs. 5.6, 5.7). Also, the percent overestimation of duplications and losses decreased with the amount of incomplete sampling (Fig. 5.3(b-c)). For example, for post-sequence (MR) analysis, Dup-loss estimated 1.08% more duplications for 0-25% deletion case and 0.14% fewer duplications for 50-75% deletion case (Fig. 5.3(b)).

The duplication and loss estimates decrease with the size of the species tree in both the 400

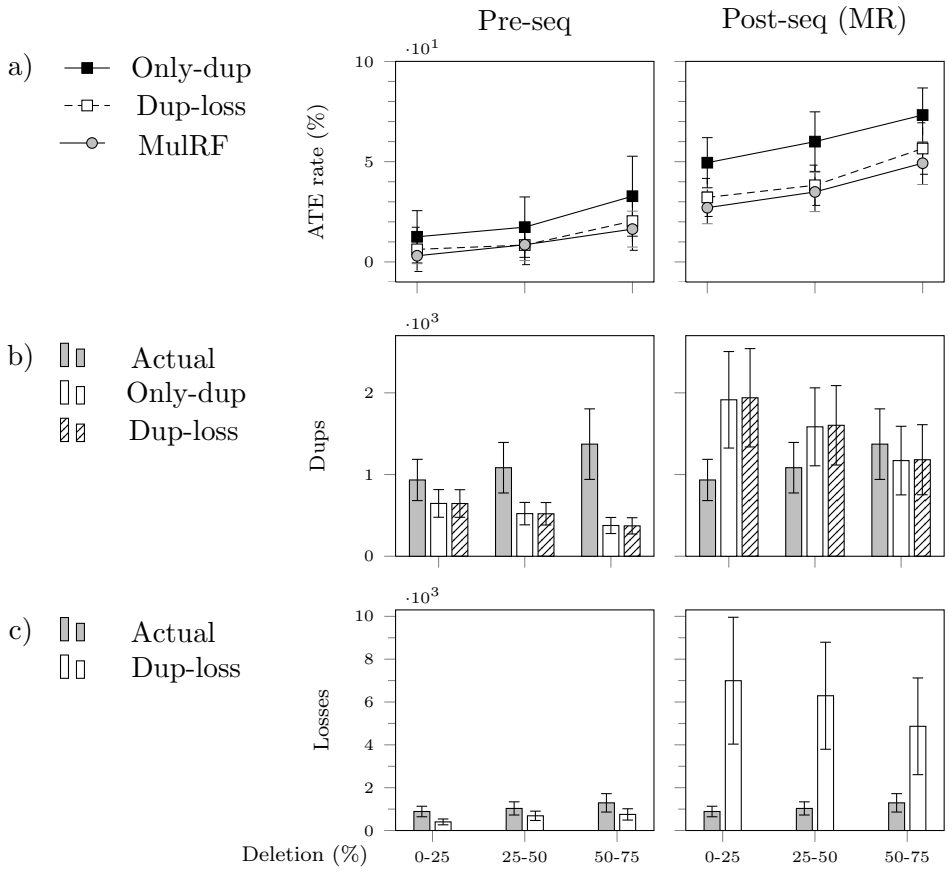

Figure 5.3 Results from the incomplete sampling experiment: a) ATE rates of the estimated species trees by Only-dup, Dup-loss, and MulRF methods, b) estimated duplications by Only-dup and Dup-loss with actual number of duplications in the gene trees, and c) estimated losses by Dup-loss with actual number of losses in the gene trees, in the pre-sequence and post-sequence (MR) analyses. Mean and standard error bars are shown.

Table 5.1 Average running time for Only-dup, Dup-loss, and MulRF methods in 400 gene trees experiment; times are given in hours:minutes:seconds.

Model Cond.		**Pre-seq**			**Post-seq (UR)**			**Post-seq (MR)**		
Num. Taxa	**D/L rate**	**Only-dup**	**Dup-loss**	**MulRF**	**Only-dup**	**Dup-loss**	**MulRF**	**Only-dup**	**Dup-loss**	**MulRF**
50	0.002	0:00:00	0:00:02	0:00:05	0:00:47	0:02:18	0:00:20	0:00:01	0:00:04	0:00:17
	0.004	0:00:01	0:00:02	0:00:05	0:00:50	0:02:27	0:00:26	0:00:01	0:00:05	0:00:23
	0.008	0:00:00	0:00:02	0:00:07	0:01:02	0:02:50	0:00:31	0:00:01	0:00:05	0:00:24
100	0.002	0:00:03	0:00:11	0:00:20	0:06:29	0:17:48	0:02:02	0:00:12	0:00:49	0:01:45
	0.004	0:00:03	0:00:12	0:00:31	0:06:29	0:16:47	0:02:49	0:00:12	0:00:50	0:02:07
	0.008	0:00:03	0:00:14	0:01:04	0:09:08	0:23:32	0:03:41	0:00:03	0:00:53	0:03:03
250	0.002	0:00:29	0:02:05	0:02:31	1:13:21	2:53:28	0:18:30	0:03:05	0:12:53	0:16:09
	0.004	0:00:29	0:02:17	0:06:15	1:38:10	3:08:15	0:21:05	0:03:07	0:13:07	0:19:19
	0.008	0:00:30	0:02:46	0:11:02	1:39:05	3:20:45	0:20:06	0:02:53	0:11:06	0:17:11
500	0.002	0:02:33	0:17:22	0:18:25	13:19:27	21:31:30	0:18:30	0:21:33	1:45:26	0:20:29
	0.004	0:02:28	0:13:36	0:17:28	7:54:50	17:30:40	0:15:25	0:16:26	1:21:32	0:18:31
	0.008	0:02:55	0:14:32	0:17:25	7:30:36	12:56:29	0:17:23	0:20:34	1:44:35	0:26:20

and 100 gene tree experiments (Figs. 5.4-5.7). For example, for post-sequence (UR) analysis with D/L rate 0.002, Only-dup overestimates the number of duplications by 172.78% for the 50-taxon simulations, and only by 26.64% for 500-taxon simulations (Fig. 5.4).

5.3.3 Running Time

In the pre-sequence and post-sequence experiments, all three methods had reasonable running times, with only the Dup-loss method in the 500-taxon experiment exceeding an average of an hour per run (Table 6.1). Overall, all three methods had similar execution times for pre-sequence and post-sequence (MR) analysis, with Only-dup slightly fastest and MulRF slightly slowest, except in the post-sequence (MR) with 500 taxa (Table 6.1). However, in the post-sequence (UR) analysis, Only-dup and Dup-loss take up to 50 times more time to execute compare to their MR counterparts (Table 6.1). In the UR setting, Only-dup and Dup-loss reroot the ML trees and performs new SPR searches several times during a single run. This greatly increases the time to complete execution compared to the MR setting.

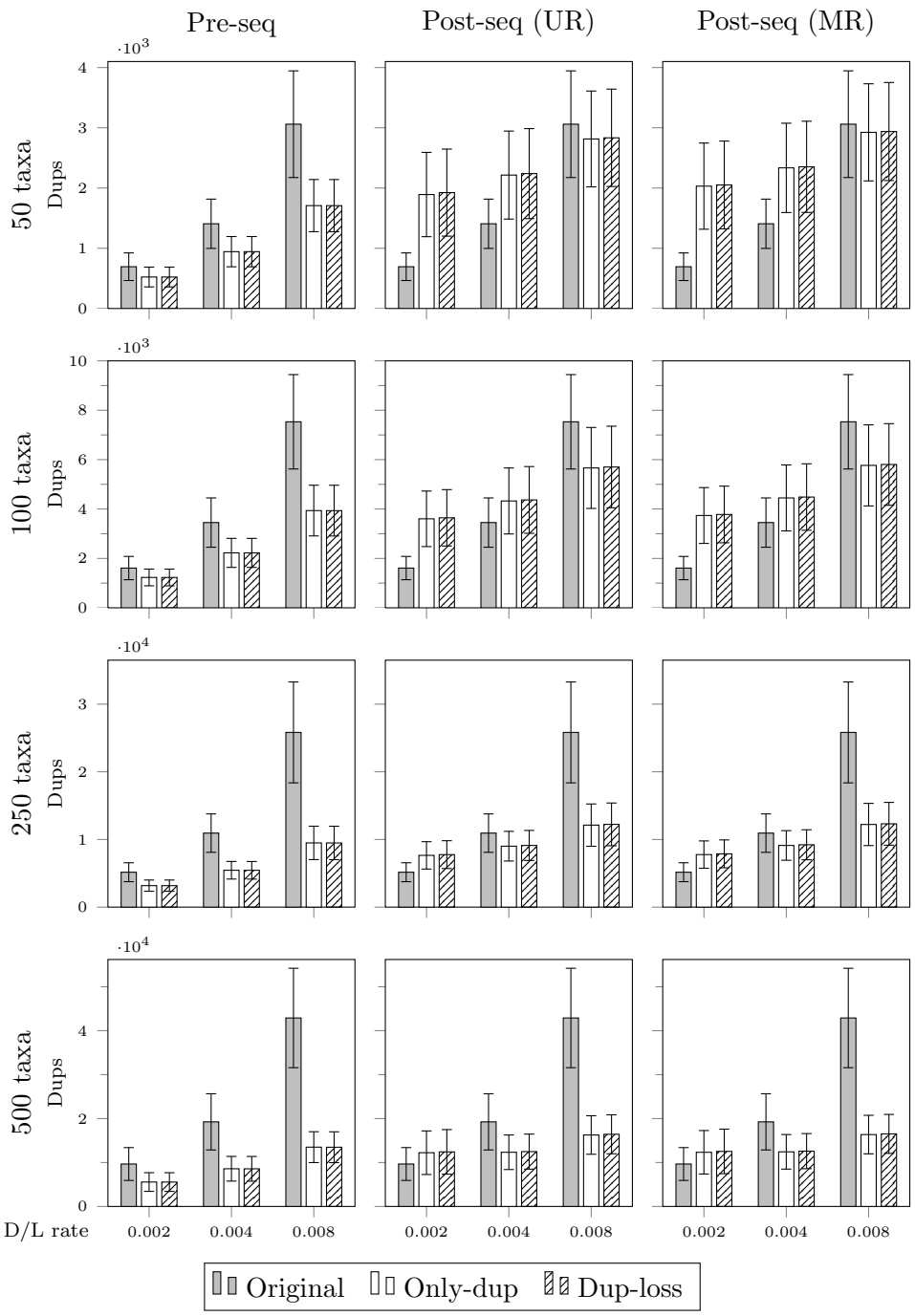

Figure 5.4 Accuracy of estimated duplications in 400 gene trees experiments. Comparison of duplications estimated by Only-dup and Dup-loss methods with the actual number of duplications in the gene trees, for differing D/L rates across 50, 100, 250, and 500 taxa model trees in the pre-sequence and post-sequence (UR and MR) analyses. Mean and standard errors are shown.

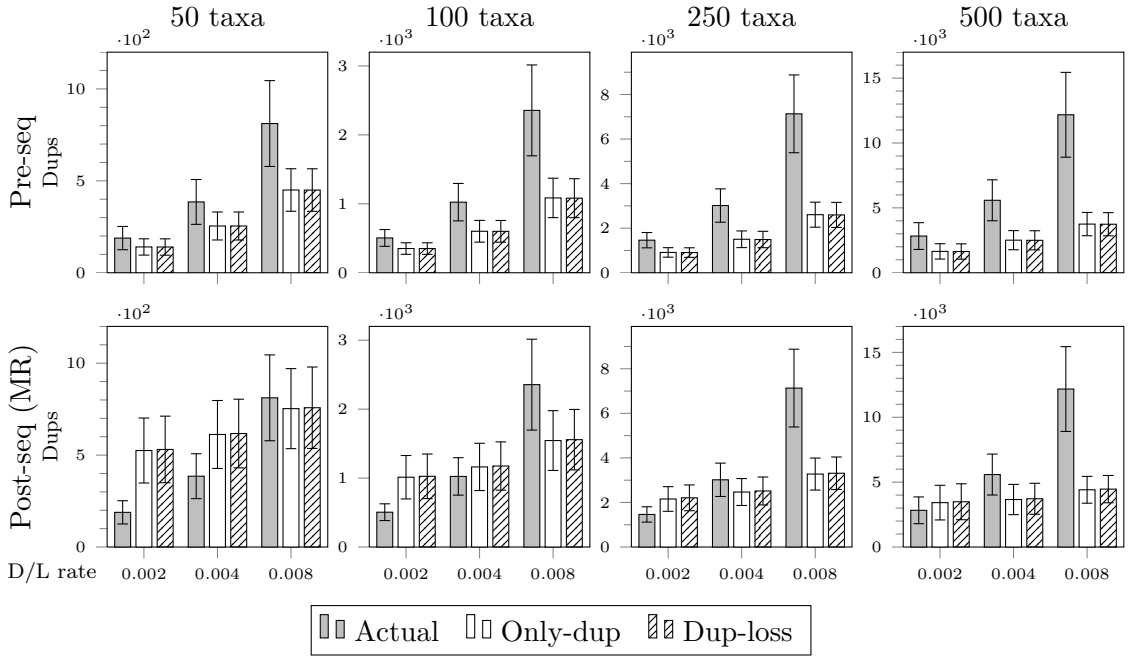

Figure 5.5 Accuracy of estimated duplications in 100 gene trees experiments. Comparing the actual duplications in the gene trees with the duplications estimated by Only-dup and Dup-loss in the pre-sequence and post-sequence (MR) analyses for differing D/L rates across 50, 100, 250, and 500 taxa model trees. Mean and standard errors are shown.

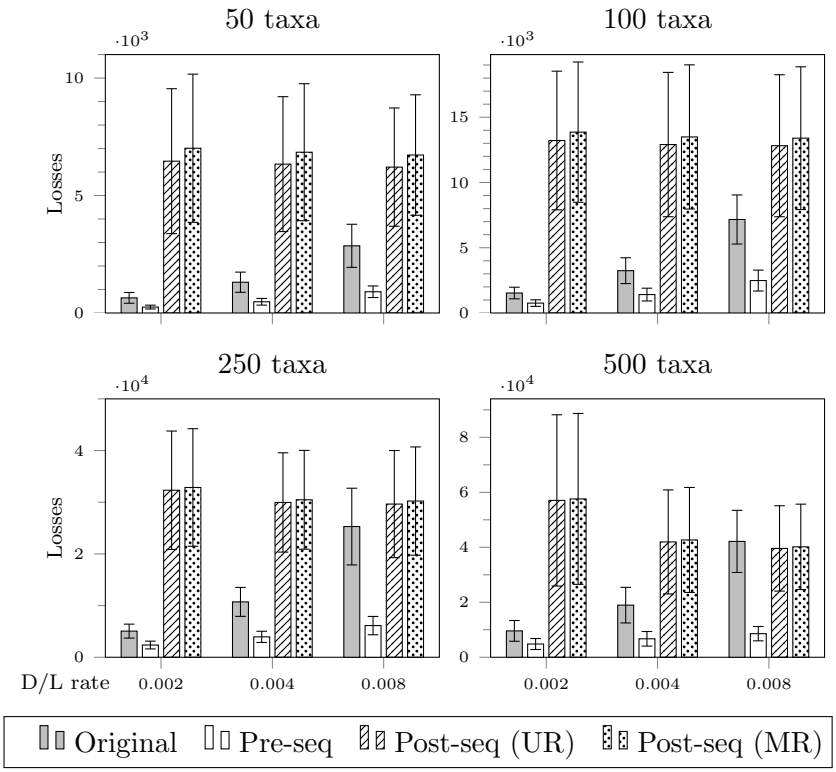

Figure 5.6 Comparison of losses estimated by Dup-loss method with the actual number of losses in the gene trees, for differing D/L rates across 50, 100, 250, and 500 taxa model trees in the pre-sequence and post-sequence (UR and MR) analyses in 400 gene trees experiments. Mean and standard errors are shown.

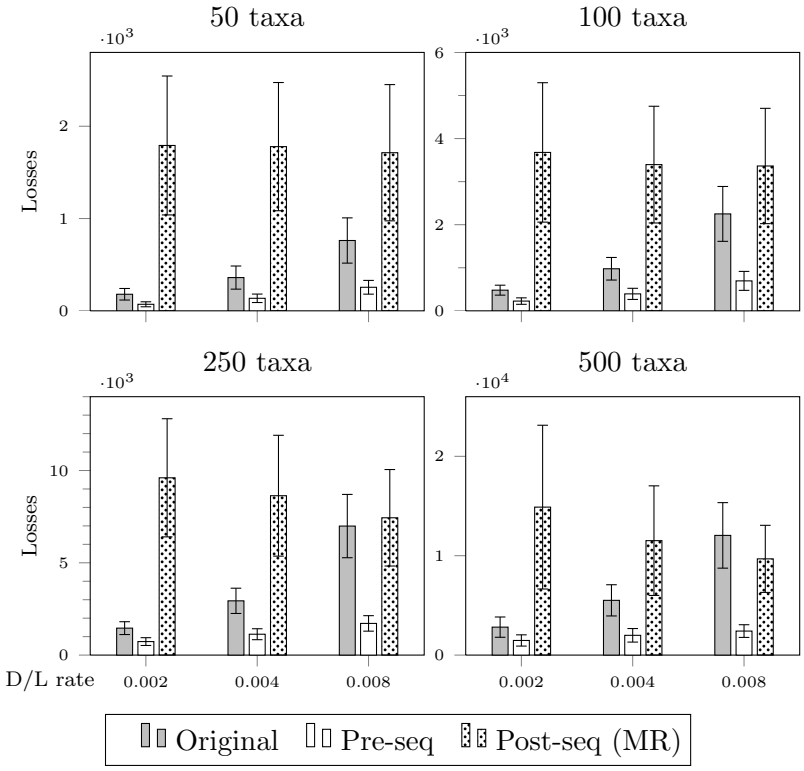

Figure 5.7 Comparing the actual losses in the gene trees with the losses estimated by Dup-loss in the pre-sequence and post-sequence (MR) analyses for differing D/L rates across 50, 100, 250, and 500 taxa model trees, in the 100 gene trees experiment. Mean and standard errors are shown.

5.4 Discussion

The simulation experiments emphasize the difficulty of constructing phylogenetic trees from gene trees with a history of duplication and loss. Even though the optimality criteria for the Only-dup and Dup-loss methods are explicitly designed to address duplications (or duplications and losses), the MulRF method, based on a mechanism-free tree distance metric, outperforms Only-dup and Dup-loss in experiments with relatively small species trees (Figs. 5.1, 5.2). However, with larger species trees, Dup-loss often is more accurate than MulRF (Figs. 5.1, 5.2). It is not surprising that both error in input trees, as introduced in the post-sequence simulations, and higher rates of duplication and loss negatively affect performance of all methods; these processes create conflict between the gene tree and species tree topologies. Incomplete gene sampling also adversely affects all of the methods (Fig. 5.3(a)), as it can mask evidence of duplications or losses. (See the example in Fig. 5.8.) The relatively poor performance of all methods in some extreme simulation conditions suggests that it may be beneficial to remove genes with especially high rates of duplication and loss or low sampling prior to phylogenetic analyses. Alternately, increasing the number of high quality input gene trees can ameliorate phylogenetic error.

In all simulation experiments, GTP using a duplication-only reconciliation cost (Only-dup; Figs. 5.1, 5.2) performs poorly compared to the other methods. Several studies have suggested that using an Only-dup cost function may be more appropriate than using the Dup-loss cost function when the input gene trees have incomplete gene sampling (e.g, Cotton and Page (2003); Burleigh et al. (2010)). In these cases, it can be impossible to distinguish gene losses from unsampled genes, and thus, estimates of gene loss may be extremely unreliable and will not represent the biological cost. Our sampling experiment suggests that this argument is unsubstantiated (Fig. 5.3); Dup-loss always outperforms Only-dup, even when only 25% of the sequences are present in the gene trees. Published analyses using Only-dup have produced credible species trees, but, based on our simulations, this is likely to occur only when the input gene trees have either very low rates of duplication and loss or when there are an enormous number of input gene trees (e.g., Burleigh et al. (2010)). In spite of its generally poor performance,

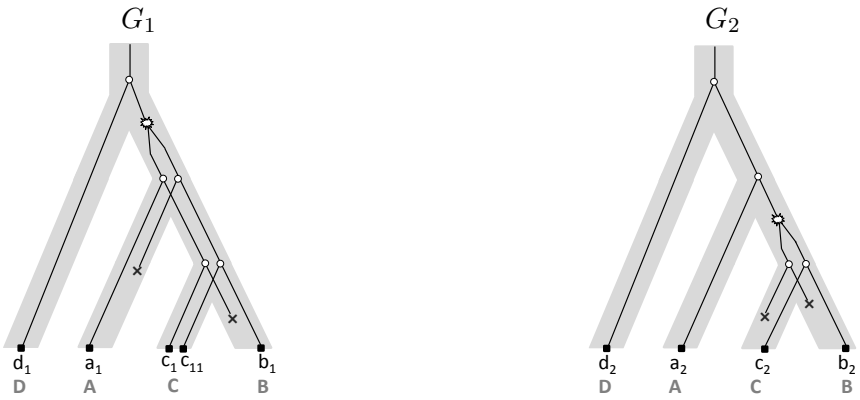

Figure 5.8 An example showing the negative effect of inadequate gene sampling. Two gene trees G_1 and G_2 are evolved over a species tree; circle, explosion, and cross signs represent speciation, duplication, and loss (or incomplete sampling) of corresponding genes, respectively. Both G_1 and G_2 are conflicting but error free. For G_1 and G_2 as input both MulRF and Only-dup estimate the right species tree, i.e., identical to G_2 in topology. Further, if the gene sequence c_{11} had not been sampled for G_1, both MulRF and Only-dup would have estimated a species tree of topology identical to than G_1 or G_2.

one advantage of Only-dup is its speed. Unlike Dup-loss and MulRF, there exists efficient and apparently effective heuristics for Only-dup that can infer species trees with 100,000 taxa in reasonable time (Wehe and Burleigh (2010)). Thus, Only-dup may still be useful for obtaining quick species tree estimates from extraordinarily large data sets.

In general, estimates of duplications or duplications and losses from Only-dup and Dup-loss, respectively, had high amounts of error. Several studies have noted that error in gene tree topologies can greatly inflate estimates of duplications (e.g., Hahn (2007); Rasmussen and Kellis (2011)). What was unexpected was the high error in duplication and loss estimates when the gene tree topologies were correct and the many situations in which GTP underestimated duplications and losses (Figs. 5.4 - 5.7).

The underestimates of duplications in the pre-sequence analyses, which used the actual gene trees from the simulations, likely are due to the inability of the GTP methods to observe duplications in a gene tree when none of the leaves under a child of duplication node are present, due to either losses or incomplete sampling (Figs. 5.4, 5.5, 5.3(b)). There are more of these "missed duplications" as the D/L rate is increased or the amount of gene sampling is reduced (Figs. 5.4, 5.5, 5.3(b)). Losses are similarly underestimated under the same conditions

(Figs. 5.6, 5.7, 5.3(c)). Missing data can lead to overestimates or underestimates of losses. Multiple losses in a subtree are observed as a single loss if the other leaves in the subtree are not sampled. In contrast, missing sequences can increase the number of perceived losses, leading to overestimation of losses. Overall in our experiments losses appear to be missed more than overcounted, and the impact of missed losses is more evident in simulations of larger trees (Figs. 5.6, 5.7, 5.3(c)).

In the post-sequence analyses for D/L rate 0.002, overestimates of duplications (by both Only-dup and Dup-loss) are likely due to errors in the gene tree topology that are interpreted as duplications (Figs. 5.4, 5.5, 5.3(b)). These "mistaken duplications" do not increase with the increasing D/L rate, but the missed duplications do. Eventually with higher D/L rates, missed duplications become more common than mistaken duplications, and consequently, duplications are overestimated when there is a low D/L rate and underestimated when there is a high D/L rate. Dup-loss overestimates losses (Figs. 5.6, 5.7, 5.3(c)) as a result of incomplete sampling and post-sequence gene tree errors, and underestimates losses due to missed losses phenomenon (explained above). As the D/L rate increases, the relative effect of missed losses becomes greater.

Whether the duplication or loss cost is over- or underestimated, it often differs greatly from the actual biological cost. The duplication or loss costs are likely to be even less accurate in analyses of real data, in which gene topologies may be further confounded by processes such as incomplete lineage sorting, recombination, and lateral transfer or reticulate evolution. Still, this does not necessarily mean that the GTP methods will produce inaccurate species trees; in fact, there does not appear to be a direct relationship between accuracy of duplication or loss estimates and species tree estimates in the simulation experiments. However, it does suggest that the (often good) performance of the GTP methods is due to the suitability of the duplication or duplication and loss cost as tree distance metrics and not the accuracy with which they reflect actual biological costs. If the duplication or duplication and loss costs do not represent the historical processes of gene evolution, it is natural to explore the performance of other tree metrics that do not purport to represent a biological cost or process, like the RF distance.

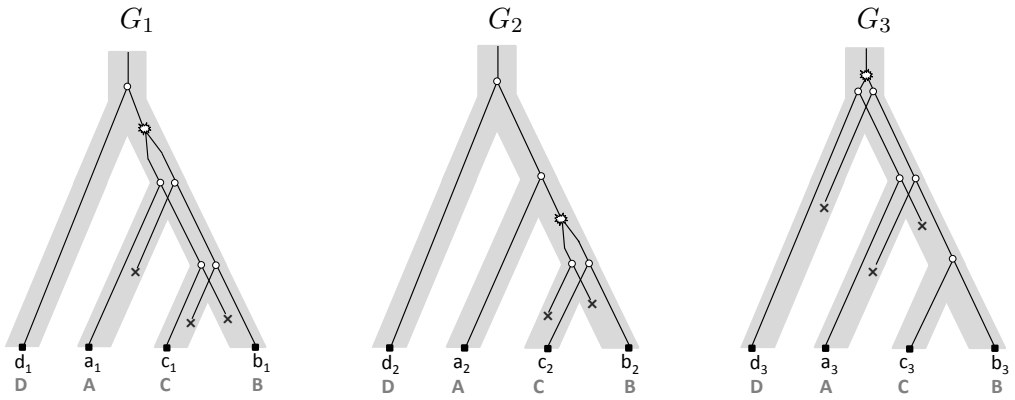

Figure 5.9 Three gene trees G_1, G_2, and G_3 evolved over a species tree; circle, explosion, and cross signs represent speciation, duplication, and loss (or incomplete sampling) of corresponding genes, respectively. Observe that the gene trees are conflicting but error free. When only G_1 and G_2 are the inputs to Only-dup and MulRF, both the methods estimate the species tree of topology identical to G_1 or G_2. After including G_3 in the input gene trees, while MulRF estimates the right species tree, i.e., identical to G_2 or G_3 in topology, Only-dup's output is same as before. Thus the additional input gene tree helps MulRF to estimate the right (unrooted) species tree, but Only-dup keeps struggling due to reliance on the rootings of the input gene trees.

One reason MulRF may outperform GTP is due to the benefits of using an unrooted tree distance metric (e.g., the metric in MulRF) instead of rooted metrics (e.g., the metrics in Only-dup and Dup-loss). The reliance of GTP on the rooting of the input gene tree can disguise the topological similarities. (See Fig. 5.9.) There has been recent work to develop GTP methods for unrooted trees (e.g., Yu et al. (2011); Górecki et al. (2012)), and rooted GTP methods have the added benefit of inferring rooted species trees, which cannot be done with MulRF (e.g., Katz et al. (2012)). Still, uncertainty and error in the root of gene trees presents a potential liability and computational challenge to GTP that can be avoided using a method based on unrooted metrics.

While the MulRF method can be effective for inferring species trees from multi-copy gene trees, other tree distance metrics, like the quartet distance, could be similarly extended to multilabeled trees and used for species tree inference. Future work should evaluate the suitability of different distance metrics for reconciling gene trees under different error models and evolutionary scenarios. The simulations demonstrate that the MulRF method often works well under models of duplication and loss, and it may also be robust to low levels of lateral gene

transfer (Chaudhary et al. (2012c), in review). However, gene tree branch swaps between distantly related taxa, as may occur in lateral gene transfer, may impact the RF distance much more than other distance metrics like a quartets distance (e.g., Ge et al. (2005)). In such cases a tree building method based on the RF distance may not be appropriate.

The application of generic (non-biological) distances to multi-copy gene reconciliation problems also suggests a variety of new applications unexplored in this study. For example, the MulRF method presents a natural approach to synthesize multi-copy gene tree data with trees built from non-molecular data (e.g., phenomic data or taxonomic trees) in a single, comprehensive phylogenetic analysis. Such analyses are possible with GTP, but it is difficult to justify using an objective based on duplications or losses when reconciling non-molecular trees. Also, the leaves of the gene trees could be represent geographic areas or host species in order to identify dominant patterns of biogeography or co-speciation (e.g., Page (1988); Ganapathy et al. (2006)). Thus, the MulRF method potentially may provide a flexible and computationally feasible approach to address large-scale evolutionary processes.

CHAPTER 6. Efficient Error Correction Algorithms for Gene Tree - Species Tree Reconciliation

6.1 Introduction

The most commonly used and computationally feasible approach to Gene tree - species tree (GT-ST) reconciliation is gene tree parsimony, which seeks to infer the fewest evolutionary events (e.g., duplication, loss, coalescence, or lateral gene transfer) needed to reconcile a gene tree and species tree topology (Maddison, 1997). This approach also can be extended to infer species phylogenies, finding the species tree that implies the fewest evolutionary events implied by the gene trees (e.g., Goodman et al. (1979); Guigó et al. (1996); Slowinski et al. (1997)). However, the gene trees often are estimated using heuristic methods from short sequence alignments, and consequently, there is often much error in the estimated gene tree topologies. Error in the gene trees creates more GT-ST incongruence and can radically affect GT-ST reconciliation analyses, implying far more duplications, duplications and losses, or deep coalescence events than actually exist. For example, Rasmussen and Kellis (2011) estimated that error in gene tree reconstruction can lead to $2-3$ fold overestimates of gene duplications and losses. Gene tree error also can erroneously imply large numbers of duplications near the root of the species tree (Burleigh et al. (2009); Hahn (2007)), and it can mislead gene tree parsimony phylogenetic analyses (e.g., Burleigh et al. (2011); Huang and Knowles (2009); Sanderson and McMahon (2007a)).

Several approaches have been proposed to address gene tree error in GT-ST reconciliation. First, questionable nodes in a gene tree or nodes with low support may be collapsed prior to gene tree reconciliation, and the resulting non-binary gene trees may be reconciled with species trees (Berglund-Sonnhammer et al. (2006); Vernot et al. (2007); Yu et al. (2011)). Similarly,

GT-ST reconciliations can use a distribution of gene tree topologies, such as bootstrap gene trees, rather than a single gene tree estimate (Burleigh et al. (2009); Cotton and Page (2002); Joly and Bruneau (2009)). Both of these approaches may help account for stochastic error and uncertainty in gene tree topologies, but they do not explicitly confront gene tree error. Methods also exist to simultaneously infer the gene tree topology and the gene tree reconciliation with a known species tree (Arvestad et al. (2004); Rasmussen and Kellis (2011)). While these sophisticated statistical approaches appear very promising, they are computationally intensive, and it is unclear if they will be tractable for large-scale analyses. Another, perhaps a more computationally feasible, approach is to allow a limited number of local rearrangements in the gene tree topology if they reduced the reconciliation cost (Chen et al. (2000); Durand et al. (2006)).

Previously (Chen et al. (2000); Durand et al. (2006)) described a method to allow NNI-branch swaps on selected branches of a gene tree to reduce the reconciliation cost. Following (Chen et al. (2000); Durand et al. (2006)), we address gene tree error in the reconciliation process by assuming that the correct gene tree can be found in a particular neighborhood of the given gene tree. We describe this approach for the gene duplication, duplication and loss, and deep coalescence models, which identify the fewest respective events implied from a given gene tree and given species tree. This neighborhood consists of all trees that are within one edit operation of the gene tree. While (Chen et al. (2000); Durand et al. (2006)) use Nearest Neighbor Interchange (NNI) edit operations to define the neighborhood, we use the standard tree edit operations SPR (Allen and Steel (2001); Bordewich and Semple (2004)) and TBR (Allen and Steel, 2001), which significantly extend upon the search space of the NNI neighborhood. The SPR and TBR local search problems find a tree in the SPR and TBR neighborhood of a given gene tree, respectively, that has the smallest reconciliation cost when reconciled with a given species tree. Using the algorithm by Zhang (1997) the best known (naïve) runtimes are $O(n^3)$ for the SPR local search problem and $O(n^4)$ for the TBR local search problem, where n is the number of taxa in the given gene tree. These runtimes typically are prohibitively long for the computation of larger GT-ST reconciliations. We improve on these solutions by a factor of n for the SPR local search problem and a factor of n^2 for the TBR

local search problem. This makes the local search under the TBR edit operation as efficient as under the SPR edit operation, and it provides a high-speed gene tree error-correction protocol that is computationally feasible for large-scale genomic data sets.

We also evaluated the performance of our algorithms using the implementation of SPR based local search algorithms. Note, that the SPR neighborhood is properly contained in the TBR neighborhood for any given tree. Thus the performance of the SPR based program provides a conservative estimate of the performance of the TBR based program. We test our programs on a collection of 106 yeast gene trees, some of which contain hundreds of leaves, and we demonstrate how it can be easily incorporated into large-scale gene tree parsimony phylogenetic analyses.

6.2 Preliminaries

6.2.1 The Reconciliation Cost Models

A *species tree* is a phylogenetic tree in which each leaf represents a species, whereas in a *gene tree* each leaf represents a sequence, encoding one gene (or gene family), for a given set of species. We assume that each leaf of the gene tree is labeled with the species from which that gene was sampled. Let G be a gene tree and S a species tree.

The *leaf-mapping* $\mathcal{L}_{G,S}\colon \mathcal{L}(G) \to \mathcal{L}(S)$ is a surjection that maps each leaf $g \in \mathcal{L}(G)$ to that unique leaf $s \in \mathcal{L}(S)$ which has the same label as g. The extension $\mathcal{M}_{G,S}\colon V(G) \to V(S)$ is the mapping defined by $\mathcal{M}_{G,S}(g) := \mathrm{LCA}(\mathcal{L}_{G,S}(\mathcal{L}(G_g)))$. For convenience, we write $\mathcal{M}(g)$ instead of $\mathcal{M}_{G,S}(g)$ when G and S are clear from the context.

Given trees G and S, we say that G is *comparable* to S if a leaf-mapping $\mathcal{L}_{G,S}(g)$ is well defined.

Definition 4 (Duplication cost).

- The *duplication cost* from $g \in V(G)$ to S, $\mathcal{C}_D(G,S,g) := \begin{cases} 1, & \text{if } \mathcal{M}(g) \in \mathcal{M}(Ch(g)); \\ 0, & \text{otherwise.} \end{cases}$
- The *duplication cost* from G to S, $\mathcal{C}_D(G,S) := \sum_{g \in I(G)} \mathcal{C}_D(G,S,g)$.

Definition 5 (**Duplication-loss cost**)**.**

- The *loss cost* from $g \in V(G)$ to S,
  $$\mathcal{C}_L(G,S,g) := \begin{cases} 0, & \text{if } \forall h \in Ch(g)\colon \mathcal{M}(g) = \mathcal{M}(h); \\ \sum_{h \in Ch(g)} |d_S(\mathcal{M}(g),\mathcal{M}(h)) - 1|, & \text{otherwise.} \end{cases}$$
- The *duplication-loss cost* from G to S, $\mathcal{C}_{DL}(G,S) := \sum_{g \in I(G)}(\mathcal{C}_D(G,S,g) + \mathcal{C}_L(G,S,g))$.

Definition 6 (**Deep coalescence cost**)**.**

- The *number of lineages* from $g \in V(G)$ to $h \in Ch(g)$ in S,
  $\mathcal{C}_{DC}(G,S,g) := \sum_{h \in Ch(g)} d_S(\mathcal{M}(g),\mathcal{M}(h))$.
- The *deep coalescence cost* from G to S, $\mathcal{C}_{DC}(G,S) := \sum_{g \in I(G)} \mathcal{C}_{DC}(G,S,g) - |E(S)|$.

The reconciliation cost are based on the models of gene duplication (Page (1994); Eulenstein (1998)), duplication-loss (Zhang, 1997), and deep coalescence (Zhang, 1997).

6.2.2 The error-correction problems

Here we give definitions for rooted tree rearrangement operations TBR (Allen and Steel, 2001) and SPR (Allen and Steel (2001); Bordewich and Semple (2004)), and then formulate the Error-Correction problems that were motivated in the introduction.

Definition 7 (Tree Bisection and Reconnection (TBR))**.** Let T be a tree. For this definition, we regard the *planted tree* $Pl(T)$ as the tree obtained from adding the *root edge* $\{r, rt(T)\}$ to $E(T)$, where $r \notin V(T)$. Let $e := (u,v) \in E(T)$, and X and Y be the connected components that are obtained by removing edge e from T such that $v \in X$ and $u \in Y$. We define $\text{TBR}_T(v,x,y)$ for $x \in X$ and $y \in Y$ to be the tree that is obtained from $Pl(T)$ by first deleting edge e, and then adjoining a new edge f between X and Y as follows:

1. If $x \neq rt(X)$ then suppress $rt(X)$ and create a new root by subdividing edge $(Pa(x), x)$.
2. Subdivide edges $(Pa(y), y)$ by introducing a new vertex y'.
3. Re-connect components X and Y by adding edge $f = (y', rt(x)$.

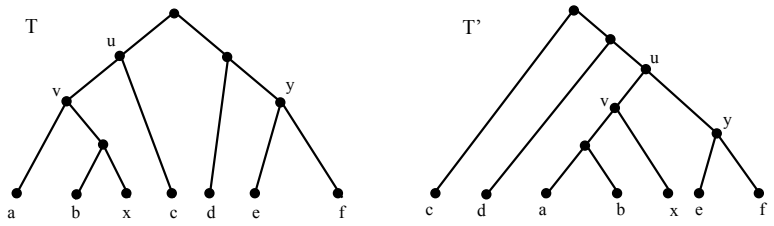

Figure 6.1 An TBR operation. Tree $T' = \text{TBR}_T(v, x, y)$ results from T after performing single TBR operation.

4. Suppress the vertex u, and rename vertex y' as u.

5. Contract the root edge.

We say that, the tree $TBR_T(v, x, y)$ is obtained from T by a *tree bisection and reconnection (TBR)* operation that bisects the tree T into the components X and Y, and reconnects them above the nodes x and y. (See Fig. 6.1) We define the following *neighborhoods* for the TBR operation:

1. $\text{TBR}_G(v, x) := \cup_{y \in Y} \text{TBR}_G(v, x, y)$

2. $\text{TBR}_G(v) := \cup_{x \in X} \text{TBR}_G(v, x)$

3. $\text{TBR}_G := \cup_{(u,v) \in E(G)} \text{TBR}_G(v)$

Definition 8 (Subtree Prune and Regrafting (SPR))**.** The SPR operation is defined as a special case of the TBR operation. Let $e := (u, v) \in E(T)$, and X and Y be the connected components that are obtained by removing edge e from T such that $v \in X$ and $u \in Y$. We define $\text{SPR}_T(v, y)$ for $y \in Y$ to be $TBR_T(v, v, y)$. We say that the tree $\text{SPR}_T(v, y)$ is obtained from T by performing subtree prune and regraft (SPR) operation that prunes subtree T_v and regrafts it above y. (See Fig. 6.2(a).)

We define the following *neighborhoods* for the SPR operation:

1. $\text{SPR}_G(v) := \cup_{y \in Y} \text{SPR}_G(v, y)$

2. $\text{SPR}_G := \cup_{(u,v) \in E(G)} \text{SPR}_G(v)$

We now state the SPR based error-correction problems for duplication (D), duplication-loss (DL), and deep coalescence (DC). Let $\Gamma \in \{\text{D, DL, DC}\}$.

Problem 5 (SPR based error-correction for Γ (SEC-Γ)).
Instance: A gene tree G and a species tree S.
Find: A gene tree $G^* \in \text{SPR}_G$ such that $\mathcal{C}_\Gamma(G^*, S) = \min_{G' \in \text{SPR}_G} \mathcal{C}_\Gamma(G', S)$.

The TBR based error-correction for Γ (TEC-Γ) problems are defined analogously to the SPR based error-correction for Γ (SEC-Γ) problems.

6.3 Solving the SEC-Γ problems

In this section we study the SPR based error-correction problems, for duplication (D), duplication-loss (DL), and deep coalescence (DC), in more detail. Our efficient solution for these problems are based on solving restricted versions of these problems efficiently. For each $\Gamma \in \{D, DL, DC\}$ we first define a restricted version of the SEC-Γ problem, which we call the restricted SPR based error-correction for the Γ (R-SEC-Γ) problem.

Problem 6 (Restricted SPR based error-correction for Γ (R-SEC-Γ)).
Instance: A gene tree G, a species tree S, and $v \in V(G)$.
Find: A gene tree $G^* \in \text{SPR}_G(v)$ such that $\mathcal{C}_\Gamma(G^*, S) = \min_{G' \in \text{SPR}_G(v)} \mathcal{C}_\Gamma(G', S)$.

Observation 2. *Let $\Gamma \in \{D, DL, DC\}$. Given a gene tree G and a species tree S, the SEC-Γ problem can be solved as follows: (i) solve the R-SEC-Γ problem for every $v \in V(G)$ where $v \neq rt(G)$, (ii) under all solutions found return a minimum scoring gene tree G^*.*

Naïvely, the R-SEC-Γ problem can be solved in $\Theta(n^2)$ time by computing the cost $\mathcal{C}_\Gamma(G', S)$ for each $G' \in \text{SPR}_G(v)$. The cost for a given gene and species tree can be computed in $\Theta(n)$ time (Zhang, 1997). We introduce a novel algorithm for the R-SEC-Γ problem that improves by a factor of n on the naïve solution. This speedup is achieved by semi-ordering the trees in $SPR_G(v)$, for each $v \in V(G)$, such that the score-difference of any two consecutive trees in this order can be computed in constant time.

Ordering the trees in $\mathbf{SPR}_G(v)$

Consider a graph on trees in $SPR_G(v)$, in which every two adjacent trees are one NNI (Allen and Steel, 2001) operation apart. We show that such a graph is a rooted full binary tree, after providing necessary definitions.

Definition 9 (Nearest Neighbor Interchange (NNI))**.** We define the NNI operation as a special case of the SPR operation. Let $e \in E(T)$ where $e := (u, v)$, and X and Y be the connected components that are obtained by removing edge e from T such that $v \in X$ and $u \in Y$. We define $\mathrm{NNI}_T(v)$ to be $\mathrm{SPR}_T(v, y)$ for $y := Pa(u)$, and say that $\mathrm{NNI}_T(v)$ is obtained from T by performing nearest neighbor interchange (NNI) operation that prunes subtree T_v and regrafts it above the parent of v's parent. (See Fig. 6.2(b).)

Definition 10 (NNI distance)**.** Let the *NNI-distance*, denoted as $d_{\mathrm{NNI}}(T_1, T_2)$, between two trees T_1 and T_2 over n taxa be the minimum number of NNI operations required to transform T_1 into T_2.

Definition 11 (NNI-adjacency graph)**.** The *NNI-adjacency graph*, denoted as $\mathcal{X} = (V, E)$, is the graph where $V = \mathrm{SPR}_G(v)$ and $\{T_1, T_2\} \in E$ if $d_{\mathrm{NNI}}(T_1, T_2) = 1$.

Lemma 14. *$\mathcal{X}$ is a tree.*

Proof. We prove it by showing that there exists a unique path between every two vertices in $\mathcal{X}$. Let $G', G'' \in V(\mathcal{X})$, thus $G', G'' \in \mathrm{SPR}_G(v)$. Let $G' := \mathrm{SPR}_G(v, x_1)$ and $G'' := \mathrm{SPR}_G(v, x_2)$. We use induction on $d_G(x_1, x_2)$. Let $d_G(x_1, x_2) = 1$ and assume without loss of generality that $x_2 = Pa_G(x_1)$. Thus, $G' = \mathrm{NNI}_{G''}(Sb(x_1))$. So the hypothesis holds for $d_G(x_1, x_2) = 1$. Assume now that the hypothesis is true for $d_G(x_1, x_2) \leq k$ and suppose $d_G(x_1, x_2) = k + 1$. Since G is a tree, there must be a unique path between x_1 and x_2; let y be a vertex on this path. Let $d_G(y, x_1) = 1$, and $G^n := \mathrm{SPR}_G(v, y)$. If $y = Pa_G(x_1)$, then $G^n = \mathrm{NNI}_{G'}(v)$; otherwise $G^n = \mathrm{NNI}_{G'}(Sb(y))$. Since $d_G(y, x_2) = k$, thus (by induction hypothesis) the hypothesis is valid for $d_G(x_1, x_2) = k + 1$. □

Theorem 7. *$\mathcal{X}$ is a rooted full binary tree.*

Proof. In view of Lemma 14, it suffices to show that except a unique vertex of degree 2 all other vertices in $\mathcal{X}$ are of degree 1 or 3. Let $G' \in V(\mathcal{X})$, thus $G' = \text{SPR}_G(v, y)$ for some $y \in V(G)$. There are three cases:

Case 1: y is a root. Let $y_1 \in Ch_G(y)$. Let $G^1 := \text{SPR}_G(v, y_1)$, thus $G' = \text{NNI}_{G^1}(v)$. Hence $\{G^1, G'\} \in E(\mathcal{X})$. Since $|Ch_G(y)| = 2$, G' must be a degree 2 vertex in $\mathcal{X}$.

Case 2: y is a leaf. Let $y_1 = Pa_G(y)$. Let $G^1 := \text{SPR}_G(v, y_1)$, thus $G^1 = \text{NNI}_{G'}(v)$. Hence $\{G^1, G'\} \in E(\mathcal{X})$, and consequently, G' is a degree 1 vertex in $\mathcal{X}$.

Case 3: y is an internal vertex. Let $y_1 = Pa_G(y)$ and $y_2 \in Ch_G(y)$. Let $G^1 := \text{SPR}_G(v, y_1)$, thus $G^1 = \text{NNI}_{G'}(v)$. Let $G^2 := \text{SPR}_G(v, y_2)$, thus $G' = \text{NNI}_{G^2}(v)$. Since y has one parent and two children in G, thus G' is a degree 3 vertex in $\mathcal{X}$.

This completes the proof. □

The score difference of consecutive trees in $\mathcal{X}$

To solve the R-SEC-Γ problems we traverse tree $\mathcal{X}$. Two adjacent trees in $V(\mathcal{X})$ are one NNI operation apart. We show that $\mathcal{C}_\Gamma$ score of a tree can be computed in constant time from the LCA computation of its adjacent tree.

Let $e := (G', G'')$ be an edge in $\mathcal{X}$. Let $x := Pa(v)$, $y := Sb(v)$, and $z, z' \in Ch(y)$ in G' (see Fig. 6.2(b)). Without loss of generality, let $G'' := \text{NNI}_{G'}(z)$. (Observe G'' is similar to G'_r of Fig. 6.2(b).)

Lemma 15. *$\mathcal{M}_{G'',S}(y) = \mathcal{M}_{G',S}(x)$.*

Proof. From NNI operation, $v, z' \in Ch_{G''}(x)$ and $z, x \in Ch_{G''}(y)$. Also, $G'_z \simeq G''_z$, $G'_{z'} \simeq G''_{z'}$, $G'_v \simeq G''_v$, so $Le(G''_y) = Le(G'_x)$. Thus, $\mathcal{M}_{G',S}(x) = \text{LCA}(\mathcal{L}_{G',S}(Le(G'_x))) = \text{LCA}(\mathcal{L}_{G'',S}(Le(G''_y)))$ $= \mathcal{M}_{G'',S}(y)$. □

Lemma 16. *$\mathcal{M}_{G'',S}(w) = \mathcal{M}_{G',S}(w)$, for all $w \in V(G')\backslash\{x, y\}$.*

Proof. For $g \in V(G'_v) \bigcup V(G'_z) \bigcup V(G'_{z'})$, since $G'_g \simeq G''_g$, therefore $\mathcal{M}_{G',S}(g) = \mathcal{M}_{G'',S}(g)$. Also, except for subtree G'_x, the rest of the tree remains the same in G''_x. Thus by Lemma

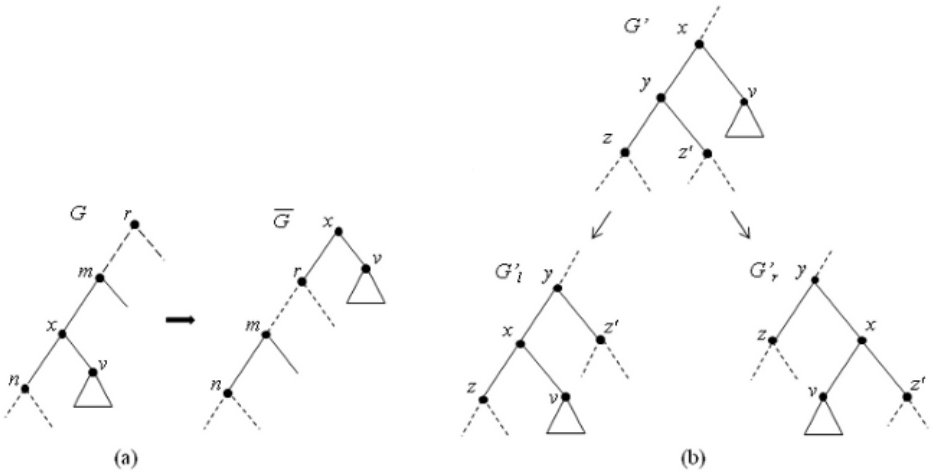

Figure 6.2 (a) The tree $\overline{G}$ is obtained from G by pruning and regrafting subtree G_v to the root of G. The vertex $x \in V(G)$ is suppressed, and the new vertex above root in $\overline{G}$ is named x. (b) Two NNI operations $\text{NNI}_{G'}(z')$ and $\text{NNI}_{G'}(z)$ produce left-child G'_l and right-child G'_r of G' in the NNI adjacency graph $\mathcal{X}$.

15, $\mathcal{M}_{G',S}(Pa_{G'}(x)) = \mathcal{M}_{G'',S}(Pa_{G''}(y))$. Inductively, $\mathcal{M}_{G',S}(g) = \mathcal{M}_{G'',S}(g)$, for all $g \in V(G')\backslash V(G'_x)$. □

Lemma 17. $\mathcal{M}_{G'',S}(x) = LCA(\mathcal{M}_{G',S}(v), \mathcal{M}_{G',S}(z'))$.

Proof. From Lemma 16, $\mathcal{M}_{G'',S}(v) = \mathcal{M}_{G',S}(v)$ and $\mathcal{M}_{G'',S}(z') = \mathcal{M}_{G',S}(z')$. Thus, $\mathcal{M}_{G'',S}(x)$ $= \text{LCA}(\mathcal{M}_{G'',S}(v), \mathcal{M}_{G'',S}(z')) = \text{LCA}(\mathcal{M}_{G',S}(v), \mathcal{M}_{G',S}(z'))$. □

Lemma 18. $\mathcal{C}_\Gamma(G'', S, g) = \mathcal{C}_\Gamma(G', S, g)$, *for all* $g \in V(G'')\backslash\{x, y\}$ *and* $\Gamma \in \{D, DL, DC\}$.

Proof. The gene duplication and loss status of a vertex, and the number of lineages from a vertex to its children in G' can change in G'' if its mapping or mapping of any of its children changes in $\mathcal{M}_{G'',S}$. From Lemma 16, and also, since $\mathcal{M}_{G'',S}(w) = \mathcal{M}_{G',S}(w)$, for $w \in Ch(Pa_{G'}(x))$, must have $\mathcal{C}_\Gamma(G'', S, Pa_{G'}(x)) = \mathcal{C}_\Gamma(G', S, Pa_{G'}(x))$. Thus the Lemma follows. □

Let $e := (G', G'') \in E(\mathcal{X})$ and $\Gamma \in$ {D, DL, DC}. We define $\Gamma_e := \mathcal{C}_\Gamma(G'', S) - \mathcal{C}_\Gamma(G', S)$ with respect to the given species tree S. Observe that this score can be negative too. We study how Γ_e can be computed efficiently for each edge e in $\mathcal{X}$.

Theorem 8. $\Gamma_e = \sum_{g \in \{x,y\}} (\mathcal{C}_\Gamma(G'', S, g) - \mathcal{C}_\Gamma(G', S, g))$.

Proof. $\Gamma_e = \mathcal{C}_\Gamma(G'', S) - \mathcal{C}_\Gamma(G', S) = \sum_{g \in V(G'')} (\mathcal{C}_\Gamma(G'', S, g) - \mathcal{C}_\Gamma(G', S, g)) = \sum_{g \in V(G'') \setminus \{x,y\}} (\mathcal{C}_\Gamma(G'', S, g) - \mathcal{C}_\Gamma(G', S, g)) + \sum_{g \in \{x,y\}} (\mathcal{C}_\Gamma(G'', S, g) - \mathcal{C}_\Gamma(G', S, g)) = \sum_{g \in \{x,y\}} (\mathcal{C}_\Gamma(G'', S, g) - \mathcal{C}_\Gamma(G', S, g))$. □

Definition 12. Let $\overline{G} := \mathrm{SPR}_G(v, rt(G))$, and let $P_{G'}$ be a path from $\overline{G}$ to G' in $\mathcal{X}$. For G', we define the *score-difference* $\Gamma_{\overline{G},G'}$ as $\Gamma_{\overline{G},G'} := \sum_{e \in E(P_{G'})} \Gamma_e$.

Theorem 9. *For given S, G, and $v \in V(G)$, the tree $G' \in V(\mathcal{X})$ is the output of a R-SEC-Γ problem iff $\Gamma_{\overline{G},G'} = min_{G'' \in V(\mathcal{X})}\ \Gamma_{\overline{G},G''}$.*

Proof. Let $\Gamma_{\overline{G},G'} = min_{G'' \in V(\mathcal{X})}\ \Gamma_{\overline{G},G''}$. We prove that G' is the output of R-SEC-Γ problem. Since $\Gamma_{\overline{G},G'} = \sum_{e \in E(P_{G'})} \Gamma_e = \Gamma(G', S) - \Gamma(\overline{G}, S)$, thus G' gives the minimum normalized $\mathcal{C}_\Gamma$ score over all trees in $V(\mathcal{X})$. Hence, G' must be the output of the R-SEC-Γ problem. The other direction follows similarly. □

The algorithm

We describe a general algorithm, called Algo-R-SEC-Γ, to solve the R-SEC-Γ problem for each $\Gamma \in \{D, DL, DC\}$. Initially Algo-R-SEC-Γ computes (i) the root vertex of the NNI-adjacency graph $\mathcal{X}$, which we call $\overline{G}$, by regrafting the subtree G_v above the root of G, (ii) the LCA mapping from $\overline{G}$ to S, and (iii) the Γ score from $\overline{G}$ to S. Then recursively Algo-R-SEC-Γ computes the LCA mapping and Γ score for every vertex G' in $\mathcal{X}$ when the LCA mapping and Γ score of its parent vertex in $\mathcal{X}$ is known. Algorithm 1 details Algo-R-SEC-Γ.

Algorithm 1 - Algo-R-SEC-Γ

Input: *A gene tree G, a species tree S, and $v \in V(G)$*

Output: *A tree $G^* \in SPR_G(v)$ such that $\mathcal{C}_\Gamma(G^*, S) = \min_{G' \in SPR_G(v)} \mathcal{C}_\Gamma(G', S)$*

01. Compute $\overline{G}$ by pruning G_v and regrafting at rt(G)

02. Compute LCA mapping $\mathcal{M}_{\overline{G},S}$

03. Call $\mathcal{C}_\Gamma(\overline{G}, S)$ = Algo-Comp-Score($\overline{G}, S, \mathcal{M}_{\overline{G},S}$)

04. Set BestTree = $\overline{G}$, BestScore = 0

05. Set $G' = \overline{G}$, $\mathcal{M}_{G',S} = \mathcal{M}_{\overline{G},S}$, $\mathcal{C}_\Gamma(G', S) = \mathcal{C}_\Gamma(\overline{G}, S)$, $\Gamma_{\overline{G},G'} = 0$

06. ***For*** *each* $k \neq rt(\overline{G}_{Sb(v)})$ *in preorder traversal of* $\overline{G}_{Sb(v)}$, ***do***

07. ***If*** *not backtracking,* ***then***

08. *Set* $x = Pa_{G'}(v)$, $y = Sb_{G'}(v)$

09. *Set* $G'' = NNI_{G'}(Sb_{G'}(k))$

10. *Set* $\mathcal{M}_{G'',S} = \mathcal{M}_{G',S}$, $\mathcal{M}_{G'',S}(y) = \mathcal{M}_{G',S}(x)$

11. $\mathcal{M}_{G'',S}(x) = LCA(\mathcal{M}_{G',S}(k), \mathcal{M}_{G',S}(v))$

12. *Call* $\Gamma_{\{G',G''\}} = \sum_{h\in\{x,y\}} Algo\text{-}G\text{-}Score(G'', S, \mathcal{M}_{G'',S}, h) - Algo\text{-}G\text{-}Score(G', S, \mathcal{M}_{G',S}, h)$

13. $\Gamma_{\overline{G},G''} = \Gamma_{\overline{G},G'} + \Gamma_{\{G',G''\}}$

14. ***If*** $\Gamma_{\overline{G},G''} <$ *BestScore,* ***then***

15. *Set BestTree* $= G''$, *BestScore* $= \Gamma_{\overline{G},G''}$

16. ***Else****,*

17. *Set* $x = Pa_{G'}(v)$, $y = Pa_{G'}(x)$

18. *Set* $G'' = NNI_{G'}(v)$

19. *Set* $\mathcal{M}_{G'',S} = \mathcal{M}_{G',S}$, $\mathcal{M}_{G'',S}(x) = \mathcal{M}_{G',S}(y)$

20. *Set* $\mathcal{M}_{G'',S}(y) = LCA(\mathcal{M}_{G',S}(Sb_{G'}(x)), \mathcal{M}_{G',S}(k))$

21. *Call* $\Gamma_{\{G'',G'\}} = \sum_{h\in\{x,y\}} Algo\text{-}G\text{-}Score(G', S, \mathcal{M}_{G',S}, h) - Algo\text{-}G\text{-}Score(G'', S, \mathcal{M}_{G'',S}, h)$

22. *Set* $\Gamma_{\overline{G},G''} = \Gamma_{\overline{G},G'} - \Gamma_{(G'',G')}$

23. *Set* $G' = G''$, $\mathcal{M}_{G',S} = \mathcal{M}_{G'',S}$, $\Gamma_{\overline{G},G'} = \Gamma_{\overline{G},G''}$

24. ***Return*** *BestTree*

Algorithm 2 - Algo-Comp-Score

Input: *A gene tree* G*, a species tree* S*, and LCA mapping* $\mathcal{M}_{G,S}$

Output: $\mathcal{C}_\Gamma(G, S)$

01. score = 0

02. ***For*** *each* $g \in I(G)$ *in preorder traversal of* G, ***do***

03. *Call score = score + Algo-G-Score*$(G, S, \mathcal{M}_{G,S}, g)$

04. ***If*** Γ *is DC,* ***then***

05. *score = score -* $|E(S)|$

06. ***Return*** *score*

Algorithm 3 - Algo-G-Score

Input: *A gene tree G, a species tree S, LCA mapping $\mathcal{M}_{G,S}$, and $g \in I(G)$*

Output: $\mathcal{C}_\Gamma(G, S, g)$

01. ***If*** Γ *is* D, ***then***

02. ***If*** $\mathcal{M}(g) \in \mathcal{M}(Ch(g))$, ***then***

03. ***Return*** *1*

04. ***ElseIf*** Γ *is* DL, ***then***

05. $ls = \sum_{h \in Ch(g)} |dp(\mathcal{M}(h)) - dp(\mathcal{M}(g)) - 1|$

06. ***If*** $\mathcal{M}(g) \in \mathcal{M}(Ch(g))$, ***then***

07. ***Return*** $ls + 1$

08. ***Else***

09. ***Return*** ls

10. ***Else*** *//*Γ *is* DC

11. ***Return*** $\sum_{h \in Ch(g)} |dp(\mathcal{M}(h)) - dp(\mathcal{M}(g))|$

Lemma 19. *The R-SEC-*Γ *problem is correctly solved by Algo-R-SEC-*Γ*.*

Proof. Lemma 14-18 and Theorem 7-9 directly imply that in order to prove the correctness of algorithm Algo-R-SEC-Γ, it is sufficient to prove that it correctly returns G' of minimum $\Gamma_{\overline{G},G'}$ among all $G' \in V(\mathcal{X})$. We will show that algorithm Algo-R-SEC-Γ accounts each $G' \in V(\mathcal{X})$, correctly computes $\Gamma_{\overline{G},G'}$ for $\Gamma \in \{D, DL, DC\}$, and returns the right G' as output.

From Definition 10, $V(\mathcal{X}) = \text{SPR}_G(v)$. In Algo-R-SEC-$\Gamma$, step 1 prunes subtree G_v and regrafts it above the root of G to create $\overline{G}$. Step 5 sets G' to $\overline{G}$. The for-loop in step 6 traverses subtree $\overline{G}_{Sb(v)}$ in preorder. For each traversed vertex $k \neq rt(\overline{G}_{Sb(v)})$, step 9 builds the tree $G'' := SPR_G(v, k)$ by applying NNI operation on the last build G'. Each for-loop iteration sets G' to the last build G'' in step 23. $\overline{G}$ and $G''s$ constitute all the trees in $\text{SPR}_G(v)$.

For $\overline{G}$, step 2 computes the LCA mapping and step 5 sets $\Gamma_{\overline{G},G'}$ to zero. Following Lemma 15-17 and Theorem 8, step 10 and 11 update the LCA of G'' and step 12 computes $\Gamma_{\{G',G''\}}$

by calling algorithm Algo-G-Score. Depending on $\Gamma \in \{D, DL, DC\}$, there are three cases:

Case 1: Γ is D. Algo-G-Score returns 1, if the vertex $g \in V(G'')$ maps to the same vertex in S as any of its children maps to, otherwise 0.

Case 2: Γ is DL. Algo-G-Score computes losses by applying the formula of Definition 4. Further, it adds 1 if there is a duplication.

Case 3: Γ is DC. Algo-G-Score, returns the number of lineages from g to each of its children $h \in Ch(g)$ in S. For each $h \in Ch(g)$, depth of $\mathcal{M}(g)$ is subtracted from depth of $\mathcal{M}(h)$ to count number of edges between $\mathcal{M}(g)$ and $\mathcal{M}(h)$.

In Algo-R-SEC-Γ, step 13 computes $\Gamma_{\overline{G},G''}$ by adding $\Gamma_{\overline{G},G'}$ and $\Gamma_{\{G',G''\}}$. When backtracking, steps 17-22 are executed to restore the right G' to compute the next unique $G'' \in Ch_{\mathcal{X}}(G)$. This ensures that the correct $\Gamma_{\overline{G},G'}$ is computed for each $G' \in V(\mathcal{X})$.

In Algo-R-SEC-Γ, step 4 sets $\overline{G}$ as the BestTree and $\Gamma_{\overline{G},\overline{G}} = 0$ as BestScore. Every time a new $G'' \in Ch_{\mathcal{X}}(G)$ is encountered, step 14 compares $\Gamma_{\overline{G},G''}$ with BestScore, and updates BestTree with G'' of the minimum $\Gamma_{\overline{G},G''}$. After the for-loop, step 24 returns the BestTree. □

Lemma 20. *The R-SEC-Γ and SEC-Γ problems can be solved in $\Theta(n)$ and $\Theta(n^2)$ time, respectively.*

Proof. We will prove that the algorithm Algo-R-SEC-Γ solves the restricted SPR based error-correction problems for each $\Gamma \in \{D, DL, DC\}$ in $\Theta(n)$ time. In Algo-R-SEC-Γ, step 1 takes constant time. Step 2 precomputes LCA values for species tree in O(n) time (Bender and Farach-Colton, 2000), and so, finds LCA mapping from $\overline{G}$ to S in $O(n)$ time in bottom-up manner. Step 3 computes the duplication, duplication-loss or deep coalescence score of $\overline{G}$ and S by calling Algo-Comp-Score. In Algo-Comp-Score, step 1 and step 2 runs for $O(1)$ and $O(n)$ time, respectively. Step 3 calls Algo-G-Score in each iteration of for-loop. Algo-G-Score runs for $O(1)$ time for $\Gamma \in \{D, DL, DC\}$.

When Γ is DC, steps 4 and 5 are further executed in Algo-Comp-Score for constant time. Thus in Algo-R-SEC-Γ, step 3 runs for $O(n)$ time. Further, steps 4 and 5 take constant time. The loop of step 6 runs for $\Theta(n)$ time. If condition of step 7 is true, steps 8-10 executes in

constant time. With precomputed LCA values from step 2, step 11 executes in constant time. Algo-G-Score runs for constant time for $\Gamma \in \{D, DL, DC\}$, and lets step 12 to execute in constant time. Further, steps 13-15 execute for constant time too. If the condition in step 7 is false, then steps 17-22 execute in constant time, similarly. Finally, step 23 runs for constant time, and hence, the R-SEC-Γ problem can be solved in $\Theta(n)$ time. From Observation 1, Algo-R-SEC-Γ is called $\Theta(n)$ time to solve SEC-Γ problem. Thus, the SEC-Γ problem can be solved in $\Theta(n^2)$ time. □

6.4 Solving the TEC-Γ problems

In this section we study the TBR based error-correction problems, for duplication (D), duplication-loss (DL), and deep coalescence (DC). More precisely, we extend our solution for the SEC-Γ problems to solve the TEC-Γ problems. A TBR operation can be viewed as an SPR operation, except that the pruned subtree can be rerooted before it is regrafted. Our speed-up for the TEC-Γ problems is achieved by observing that the Γ scores of any re-rooted pruned subtree and its remaining pruned tree are independent of each other. We define the R-TEC-Γ problems for the TEC-Γ problems, as we defined the R-SEC-Γ problems for the SEC-Γ problems. We will show that the R-TEC-Γ problems can be solved by solving two smaller problems separately and combining their solutions.

Definition 13. Let T be a tree and $x \in V(T)$. $\mathrm{RR}(T, x)$ is defined to be the tree T, if $x = rt(T)$ or $x \in Ch(rt(T))$. Otherwise, $\mathrm{RR}(T, x)$ is the tree obtained by suppressing $rt(T)$, and subdividing the edge $(Pa(x), x)$ by the new root node.

Lemma 21. *Given a tuple $\langle G, S, v\rangle$, and $G'' := TBR_G(v, x, y)$, for $x \in V(G_v)$, $y \in V(G)\backslash V(G_v)$. Then, $\mathcal{C}_\Gamma(G'', S) \leq_{G' \in TBR_G(v)} \mathcal{C}_\Gamma(G', S)$ iff $\mathcal{C}_\Gamma(RR(G_v, x), S) \leq_{x' \in V(G_v)} \mathcal{C}_\Gamma(RR(G_v, x'), S)$ and $\mathcal{C}_\Gamma(G'', S) \leq_{G' \in TBR_G(v,x)} \mathcal{C}_\Gamma(G', S)$.*

Proof. ($\Rightarrow$) Let $G^1 := TBR_G(v, x_1, y)$, for $x_1 \in V(G_v)$, and $x_1 \neq x$. Now observe that, $\forall g \in V(G)\backslash V(G_v)$, $\mathcal{C}_\Gamma(G'', S, g) = \mathcal{C}_\Gamma(G^1, S, g)$. Also, let $G^2 := TBR_G(v, x, y_1)$, for $y_1 \in V(G)\backslash V(G_v)$, and $y_1 \neq y$. Observe that, $\forall g \in V(G_v)$, $\mathcal{C}_\Gamma(G'', S, g) = \mathcal{C}_\Gamma(G^2, S, g)$. Thus, if G'' gives the minimum duplication, duplication-loss, or deep coalescence score among all trees

in $\text{TBR}_G(v)$, then the score contribution of vertices in $V(G_v)$ and $V(G)\backslash V(G_v)$ is independent. Now looking at vertices of G, the best score is achieved when G_v is rooted at x, i.e. $\mathcal{C}_\Gamma(RR(G_v, x), S) \leq_{x' \in V(G_v)} \mathcal{C}_\Gamma(RR(G_v, x'), S)$; also the best score is achieved when $RR(G_v, x)$ is regrafted at y, i.e., $\mathcal{C}_\Gamma(G'', S) \leq_{G' \in \text{TBR}_G(v,x)} \mathcal{C}_\Gamma(G', S)$. ($\Leftarrow$) This follows similarly. □

Lemma 8 implies that a tree in $\text{TBR}_G(v)$ with the minimum duplication, duplication-loss, or deep coalescence cost can be obtained by optimizing the rooting for the pruned subtree, and the regraft location, separately. A best rooting for the pruned subtree is linear time computable (Górecki and Tiuryn, 2006; Chen et al., 2000), and the solution to the R-SEC problem identifies a best regraft location in $\Theta(n)$ time. This allows to obtain a tree in $\text{TBR}_G(v)$ with the minimum duplication, duplication-loss, or deep coalescence cost by evaluating only $\Theta(n)$ trees. Thus the R-TEC-Γ problem can be solved in $\Theta(n)$ time. The TEC-Γ problem can be solved by calling the solution of R-TEC-Γ problem $\Theta(n)$ times, and Theorem 10 follows.

Theorem 10. *The TEC-Γ problem can be solved in $\Theta(n^2)$ time.*

6.5 Experimental results

We tested the performance of the gene tree rearrangement algorithms on a set of 106 gene alignments containing sequences from 8 yeast taxa from Rokas et al. (2003). There is a well accepted phylogeny for the yeast species, and the data set has been used to test algorithms for gene tree parsimony based on the deep coalescence problem (Than and Nakhleh, 2009; Bansal et al., 2010a). We constructed maximum likelihood gene trees for each gene using RAxML-VI-HPC version 7.0.4 (Stamatakis, 2006b), the gene trees were rooted with the outgroup *Candida albicans*. We used the new error correction algorithms to examine how much a single SPR rearrangement in the gene tree reduces the reconciliation cost based on deep coalescence and also gene duplications and losses. Over all genes the SPR error correction reduced the total deep coalescence cost from 151 to 53 (Table 6.1) and the duplication and loss cost from 481 to 175 (Table 6.2). Both the algorithms took only seconds to run for all 106 genes on a standard laptop.

Table 6.1 Error correction based on deep coalescence model. The number of yeast gene trees with different reconciliation costs based on the deep coalescence model both before (Original) and after (Post-Correction) the SPR error correction.

Reconciliation Cost	**Original**	**Post-Correction**
0	45	77
1	32	15
2	6	8
3	9	5
4	8	0
>4	6	1

We also implemented a protocol to use the gene rearrangement algorithm to correct for gene tree error in gene tree parsimony phylogenetic analyses. We first took a collection of input gene trees and performed a SPR species tree search using Duptree (Wehe et al., 2008), which seeks the species tree with the minimum gene duplication cost. We used the duplication only cost (instead of duplications and losses) because when there is no complete sampling of all existing genes, the loss estimates may be inflated by missing sequences. After finding the locally optimal species tree, we used our SPR gene tree rearrangement algorithm to find gene tree topologies with a lower duplication cost. We then performed another SPR species tree search using Duptree, starting from the locally optimal species tree and using the new gene tree topologies. This search strategy is similar to re-rooting protocol in Duptree, which checks for better gene tree roots after a SPR species tree search (Chang et al. (2011); Wehe et al. (2008)). We tested this protocol on data set of 6,084 genes (with a combined $81,525$ leaves) from 14 seed plant taxa. This is the same data set used by (Chang et al., 2011), except that all gene tree clades containing sequences from a single species were collapsed to a single leaf. Our original SPR tree search found a species tree with $23,500$ duplications. The SPR tree search after the gene tree rearrangements identified the same species tree, but the new gene trees had a reconciliation cost of only $18,213$. This tree search protocol took just under 4 hours on a Mac Powerbook with a 2 GHz Intel Core 2 Duo processor and 2 GB memory.

Table 6.2 Error correction based on duplication and loss model. The number of yeast gene trees with different reconciliation costs based on the duplication and loss model both before (Original) and after (Post-Correction) the SPR error correction.

Reconciliation Cost	Original	Post-Correction
0	45	77
1-5	32	15
6-10	15	13
11-15	8	0
16-20	5	1
>20	1	0

6.6 Conclusion

GT-ST reconciliation provides a powerful approach to study the patterns and processes of gene and genome evolution. Yet it can be thwarted by the error that is an inherent part of gene tree inference. Any reliable method for GT-ST reconciliation must account for gene tree error; however, any useful method also must be computationally tractable for large-scale genomic data. We introduce fast and effective algorithms to correct error in the gene trees. These algorithms, based on SPR and TBR rearrangements, greatly extend upon the range of possible errors in the gene tree from existing algorithms (Chen et al., 2000; Durand et al., 2006), while remaining fast enough to use on data sets with thousands of genes. These algorithms can correct trees based on a broad variety of evolutionary factors that can cause conflict between gene trees and species trees, including gene duplication, duplications and losses, and deep coalescence.

Our analysis on 106 yeast gene trees demonstrates that even a single SPR correction on the gene trees can radically improve upon the reconciliation cost. Our results in the yeast analysis are very similar to the 2-3 fold improvement in implied duplications and losses reported from the parametric gene tree estimation and reconciliation method of Rasmussen and Kellis (2011). However, in contrast, to this computationally complex method, the gene tree rearrangement algorithm is extremely fast, does not require assumption about the rates of duplication and loss, and is also amenable to corrections based on deep coalescence and duplications as well as duplications and losses. We do not claim that the gene correction algorithms produce a more

accurate reconciliation than these parametric methods. However, they do present an extremely fast and flexible alternative.

We also demonstrated that this error correction protocol could easily be incorporated into a gene tree parsimony phylogenetic analysis. Previous studies have emphasized that gene tree parsimony is sensitive to the topology of the input trees. For example, the species tree may differ whether the gene trees are made using parsimony or maximum likelihood (Burleigh et al. (2011); Sanderson and McMahon (2007a)). In our study, although the gene tree rearrangement did not affect the species tree inference, it did greatly reduce the gene duplication reconciliation cost.

While the results of the experiments are promising, they also suggest several directions for future research. First, further investigation is needed to characterize the effects of error on gene tree topologies. For example, it seems likely that gene tree errors may extend beyond a single SPR or TBR neighborhood. Yet, if we allow unlimited rearrangements, the gene trees will simply converge on the species tree topology. One simple improvement may be to weight the possible gene tree rearrangements based on support for different clades in the gene tree. Thus, well-supported clades may be rarely or never be subject to rearrangement, while poorly supported clades may be subject to extensive rearrangements. Finally, these approaches implicitly assume that all differences between gene trees and species trees are due to either coalescence, duplications, or duplications and losses. Future work will seek to combine these objectives and also address lateral transfer.

CHAPTER 7. NP-Completeness Proofs

7.1 Computing the RF Distance between two mul-trees is NP-complete

We reiterate our main problem again as follows:

Problem 7 (Computing RF distance between two mul-trees)**.**
Instance: Two mul-trees $\mathcal{T}$ and $\mathcal{T}'$.
Find: The minimum number of contractions and refinements necessary to transform $\mathcal{T}$ into $\mathcal{T}'$.

The NP-completeness proof relies on a reduction from the following NP-complete problem (Garey and Johnson, 1979).

Problem 8 (Exact Cover by 3-Sets (X3C))**.**
Instance: $S := \{s_1, ..., s_n\}$, where $n = 3q$, and $C := \{C_1, ..., C_m\}$ such that $C_i = \{s_{i_1}, s_{i_2}, s_{i_3}\}$.
Find: Are there exist sets $C_{i_1}, ..., C_{i_q}$ such that $\bigcup_{j=1}^{q} C_{i_j} = S$?

Note that X3C remains NP-complete (Hickey et al., 2008) even when each element of S occurs in *exactly* three subsets in C, thus $m = n = 3q$. We take this version of X3C for reduction. For a given instance of the X3C problem, we construct two mul-trees $\mathcal{T}_1$ and $\mathcal{T}_2$ such that transforming from $\mathcal{T}_1$ into $\mathcal{T}_2$ (or vice versa) requires κ (to be specified later) contractions and refinements if and only if an exact cover of S exists.

The mul-trees $\mathcal{T}_1$ and $\mathcal{T}_2$ are constructed in the following way. For each $s_i \in S$, we construct two rooted, binary trees $\mathbb{T}$ and $\mathbb{T}'$ that take a "large" number of contractions and refinements to transform into each other (see Fig. 7.1). Let k and t be two positive integers such that $k + 2 \geq n^2$ and $k + 2 = 2^t$. Tree $\mathbb{T}$ and $\mathbb{T}'$ have $k + 2$ leaves. Tree $\mathbb{T}'$ has the same topology as $\mathbb{T}$, but for each cherry (x, y) in $\mathbb{T}$, x and y are in different subtrees $\mathbb{T}'_u$ and $\mathbb{T}'_v$ in $\mathbb{T}'$, where

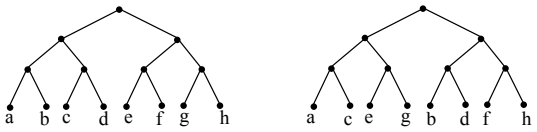

Figure 7.1 Two possible trees $\mathbb{T}$ and $\mathbb{T}'$ on 8 leaves with RF distance 12.

u and v are two children of $rt(\mathbb{T}')$. For each $s_i \in S$, corresponding trees $\mathbb{T}$ and $\mathbb{T}'$ have unique leaves.

Lemma 22. $RF(\mathbb{T}, \mathbb{T}') = 2k$.

Proof. $RF(\mathbb{T}, \mathbb{T}') = 2|\mathcal{H}(\mathbb{T}) \backslash \mathcal{H}(\mathbb{T}')|$, since $\mathbb{T}$ and $\mathbb{T}'$ are binary trees. $\mathbb{T}$ and $\mathbb{T}'$ are binary trees on $k+2$ leaves, thus $\mathcal{H}(\mathbb{T}) = \mathcal{H}(\mathbb{T}') = k$. Thus it suffices to show that no cluster in $\mathbb{T}$ matches any cluster in $\mathbb{T}'$. Let $v \in I(\mathbb{T})$, the corresponding cluster $C(v)$ contains leaves of $1 \leq p \leq (k+2)/4$ cherries. From the construction, $\mathbb{T}'$ has both leaves of each cherry in different subtrees under the root $rt(\mathbb{T}')$; thus there is no matching cluster for $C(v)$ in $\mathbb{T}'$. □

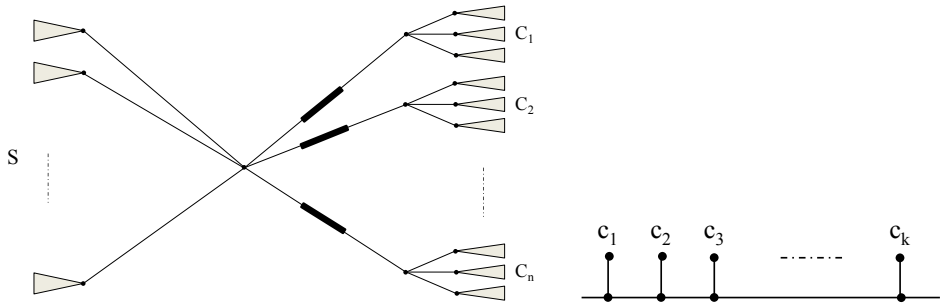

Figure 7.2 (a) Structure of mul-tree $\mathcal{T}_1$ and (b) A toll sequence of k leaves.

We are now ready for the construction of $\mathcal{T}_1$ and $\mathcal{T}_2$. Figure 7.2(a) outlines the structure of $\mathcal{T}_1$. The solid rectangles represent *toll* sequences of k uniquely labeled leaves (Fig. 7.2(b)). The left side of $\mathcal{T}_1$ has n triangles one for each of the n elements in S. Each triangle represents a tree $\mathbb{T}$ corresponding to $s_i \in S$, connecting through its root. The right side of $\mathcal{T}_1$ has n sets of 3 triangles corresponding to the subsets in C; for each subset $C_i = \{s_{i_1}, s_{i_2}, s_{i_3}\}$, the triangles represent three trees $\mathbb{T}'$s, corresponding to each s_{i_j} (for $1 \leq j \leq 3$), connected through their roots.

$\mathcal{T}_2$ has the similar structure as $\mathcal{T}_1$ except that $\mathcal{T}_2$ has tree $\mathbb{T}'$ for each $s_i \in S$ and tree $\mathbb{T}$ for each element of $C_i \in C$ (for $1 \leq i \leq n$). Thus, $\mathcal{T}_2$ has $\mathbb{T}'$s on the left side and $\mathbb{T}$s on the right side, which is opposite to what $\mathcal{T}_1$ has.

Lemma 23. *Mul-trees $\mathcal{T}_1$ and $\mathcal{T}_2$ can be constructed in polynomial time.*

Proof. Trees $\mathbb{T}$ and $\mathbb{T}'$ are rooted binary trees on $k+2$ leaves. $\mathbb{T}$ and $\mathbb{T}'$ can be constructed in polynomial time, and so the $4n$ copies of each (for $\mathcal{T}_1$ and $\mathcal{T}_2$). Further, $2n$ toll sequences (n for each $\mathcal{T}_1$ and $\mathcal{T}_2$) can be constructed in polynomial time. There are constant number of rest of the vertices in $\mathcal{T}_1$ and $\mathcal{T}_2$. Hence, the Lemma. □

Here is the connection between exactly covering S and transforming $\mathcal{T}_1$ into $\mathcal{T}_2$ by contractions and refinements: To transform $\mathcal{T}_1$ into $\mathcal{T}_2$, all we need is to convert each tree $\mathbb{T}$ on the left into $\mathbb{T}'$ and each tree $\mathbb{T}'$ on the right into $\mathbb{T}$. From Lemma 22, this costs $24qk$ contractions and refinements. A rather clever technique is to swap $3q$ $\mathbb{T}$s on the left with their counterparts on the right and to transform the remaining $6q$ $\mathbb{T}'$s on the right into $\mathbb{T}$s. If an exact cover $C_{i_1}, ..., C_{i_q}$ of S exists, we can partition the $3q$ $\mathbb{T}$s into q groups according to the cover. For each C_j $(j = i_1, ..., i_q)$ in the cover, we swap the corresponding group of trees for sequences $s_{j_1}, s_{j_2}, s_{j_3}$ with their counterparts.

Lemma 24. *All $\mathbb{T}'$s for each C_j $(j = i_1, ..., i_q)$ can be swapped with corresponding $\mathbb{T}$s by $2(k+1)$ contractions and refinements.*

Proof. Take the toll sequence corresponding to C_j and contract its $k+1$ edges; i.e., $(k-1)$ internal edges and 2 edges at both the sides of the toll sequence. Now refine it so that corresponding $\mathbb{T}$s move in C_j and $\mathbb{T}'$s stay in the left. This takes $2(k+1)$ contractions and refinements. □

From Lemma 29, if the exact cover of S exists, then $6q$ trees can be transformed by $2q(k+1)$ contractions and refinements. Remaining $6q$ $\mathbb{T}'$s can be transformed into $\mathbb{T}$s by $12qk$ contractions and refinements. Hence, we have the following lemma.

Lemma 25. *If set S has an exact cover then the RF distance between $\mathcal{T}_1$ and $\mathcal{T}_2$ is $\kappa = 2q(k+1) + 12kq$.*

If there is no exact cover of S, then either more than $6q$ trees ($\mathbb{T}$ or $\mathbb{T}'$) are transformed separately or more than q group swaps are performed. The construction guarantees that both cases will cost more than the cost of transforming ($\mathcal{T}_1$ into $\mathcal{T}_2$) in exact cover case. Hence, we conclude the following.

Theorem 11. *Set S has no exact cover if and only if the RF distance between $\mathcal{T}_1$ and $\mathcal{T}_2$ is more than $\kappa = 2q(k+1) + 12kq$.*

7.2 Tree labeling problem is NP-complete

In this Section, we prove the NP-completeness of the *tree labeling problem*: Labeling two unlabeled trees so as to minimize the RF distance between the resulting singly-labeled trees.

Let $\mathcal{T}$ be a mul-tree such that $\mathcal{T} = (T, M, \varphi)$, where T be an underlying, unrooted tree, M be the set of labels, and the surjective *labeling function* $\varphi : \mathcal{L}(T) \rightarrow M$ maps each leaf of T with a label in M. A *full differentiation* of $\mathcal{T}$ is a leaf labeled tree $\mathbf{T}$ such that T and $\mathbf{T}$ are isomorphic.

Let $\mathcal{T} = (T, M, \varphi)$ and $\mathcal{T}' = (T', M', \varphi')$ be two unrooted mul-trees. Two full differentiations $\mathbf{T}$ and $\mathbf{T}'$ of $\mathcal{T}$ and $\mathcal{T}'$, respectively, are *consistent* if for each $a \in M \cap M'$, $\tau_1(\varphi^{-1}(a)) = \tau_2(\varphi'^{-1}(a))$, where T and $\mathbf{T}$ are isomorphic under bijection $\tau_1 : V(T) \rightarrow V(\mathbf{T})$ and T' and $\mathbf{S}$ are isomorphic under bijection $\tau_2 : V(T') \rightarrow V(\mathbf{S})$. For instance, a consistent full differentiation can be obtained by relabeling each of the k copies of each leaf label a by $a_1, a_2, \ldots, a_k$ in both the mul-trees.

An unlabeled tree can also be considered as an uniform, mul-tree (i.e., a mul-tree in which the underlying set contains just one element). Consequently, the tree labeling problem can also be written in the following way:

Problem 9 (Tree Labeling)**.**
Instance: Two uniform, mul-trees $\mathcal{T}$ and $\mathcal{T}'$.
Find: The full differentiations $\mathbf{T}$ and $\mathbf{T}'$ of $\mathcal{T}$ and $\mathcal{T}'$, respectively, such that $RF(\mathbf{T}, \mathbf{T}') = \min\{RF(T, T')$: T and T' are full differentiations of $\mathcal{T}$ and $\mathcal{T}'$, respectively$\}$.

Theorem 12 (Ganapathy et al. (2006)). *Let $\mathcal{T}$ and $\mathcal{T}'$ be two mul-trees. Then, $RF(\mathcal{T},\mathcal{T}') = \min\{RF(\mathbf{T},\mathbf{T}') : \mathbf{T}$ and $\mathbf{T}'$ are mutually consistent full differentiations of $\mathcal{T}$ and $\mathcal{T}'$, respectively$\}$.*

Problem 10 (Computing RF distance between two uniform, mul-trees).
Instance: Two uniform, mul-trees $\mathcal{T}$ and $\mathcal{T}'$.
Find: The minimum number of contractions and refinements necessary to transform $\mathcal{T}$ into $\mathcal{T}'$.

Observe that Problem 10 is the special case of Problem 7, which is proved NP-complete in Section 7.1. The NP-completeness proof of Problem 10 is given later.

Theorem 13. *Problem 9 cannot be solved in polynomial time unless $P = NP$.*

Proof. Follows from the observation that Problem 10 reduces to Problem 9. More precisely, if there exists a black box that solves Problem 9 then the input of the Problem 10 can be given to it. The output contains a full differentiations of input mul-trees that minimizes the RF distance. This RF distance can be computed in linear time (Robinson and Foulds, 1981) to solve Problem 10 (from Theorem 12). □

From now on we work towards proving NP-completeness of Problem 10. Our proof relies on a reduction from the special case of the Exact Cover by 3-Sets (X3C) problem used in Section 7.1.

For a given instance of the X3C problem, we construct two uniform, mul-trees $\mathcal{T}_1$ and $\mathcal{T}_2$, such that transforming from $\mathcal{T}_1$ into $\mathcal{T}_2$ (or vice versa) requires κ (to be specified later) contractions and refinements if and only if an exact cover of S exists.

The uniform, mul-trees $\mathcal{T}_1$ and $\mathcal{T}_2$ are constructed in the following way. For each $s_i \in S$ ($1 \leq i \leq n$), we construct a complete, binary, uniform, mul-tree $\mathbb{T}^i$, and a uniform, caterpillar mul-tree $\mathbb{T}'^i$. Both mul-trees are binary, and take a "large" number of contractions and refinements to transform into each other. Let f and p be two positive integers such that $n < 2^{p-1}$ and $f = 2^p$. For each instance of Problem 7.1, p is the smallest integer that satisfies the two conditions. Both $\mathbb{T}^i$ and $\mathbb{T}'^i$ have $f + 2i$ leaves, labeled with x. $\mathbb{T}^i$ is constructed in the following way: First a rooted, perfect binary tree of height p is constructed, then left most i cherries are further extended by attaching a cherry to each of its leaves (see Fig. 7.3).

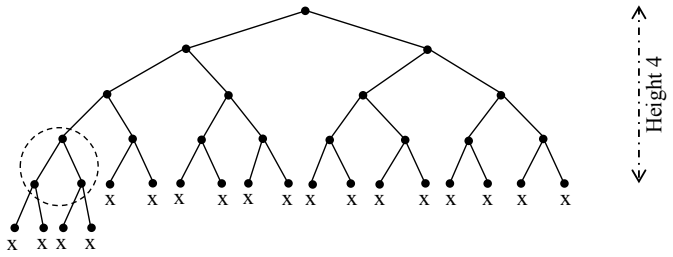

Figure 7.3 Structure of the uniform, mul-tree $\mathbb{T}^i$ for $s_i \in S$, where $i = 1$ and $n = 6$. Here, $p = 4$ and $f = 16$. The dotted circle shows the first left cherry that is extended one more level to construct $\mathbb{T}^1$ from the perfect binary mul-tree of height 4.

We are now ready for the construction of $\mathcal{T}_1$ and $\mathcal{T}_2$. Figure 7.4(a) outlines the structure of $\mathcal{T}_1$. The solid rectangles represent *toll* sequences of $2f$ leaves labeled with x (Fig. 7.4(b)). The left side of $\mathcal{T}_1$ has n triangles representing $\mathbb{T}^i$s $(1 \leq i \leq n)$, connecting through its root. The right side of $\mathcal{T}_1$ has n sets of 3 triangles corresponding to C_is $(1 \leq i \leq n)$ in C. For each $C_i = \{s_{i_1}, s_{i_2}, s_{i_3}\}$, the three dotted triangles represent $\mathbb{T}'^{i_j}$s $(1 \leq j \leq 3)$, connecting through their roots.

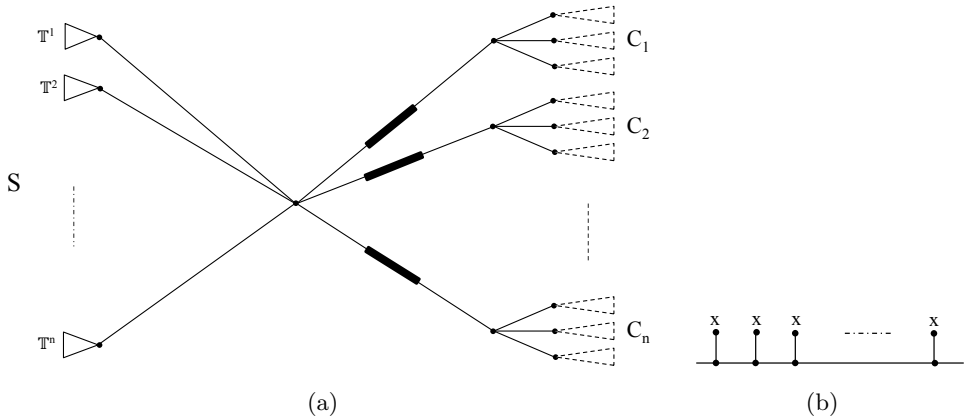

Figure 7.4 (a) Structure of uniform, mul-tree $\mathcal{T}_1$ and (b) A toll sequence of $2f$ leaves.

$\mathcal{T}_2$ has the similar structure as $\mathcal{T}_1$, except that $\mathcal{T}_2$ has $\mathbb{T}'^i$ for each $s_i \in S$ and $\mathbb{T}^{i_j}$s for each $s_{i_j} \in C_i$ (for $1 \leq i \leq n$ and $1 \leq j \leq 3$), which is opposite to what $\mathcal{T}_1$ has.

Lemma 26. *Mul-trees $\mathcal{T}_1$ and $\mathcal{T}_2$ can be constructed in polynomial time.*

Proof. Both the uniform, mul-trees $\mathbb{T}^i$ and $\mathbb{T}'^i$ have $f+2i$ leaves. Thus, they can be constructed in polynomial time, and so the $8n$ copies of each of them ($4n$ for $\mathcal{T}_1$ and $4n$ for $\mathcal{T}_2$). Further, $2n$ toll sequences (n for each $\mathcal{T}_1$ and $\mathcal{T}_2$) can be constructed in polynomial time. There are

constant number of rest of the vertices in $\mathcal{T}_1$ and $\mathcal{T}_2$. . □

Lemma 27. *$\mathbb{T}^i$ can be converted into $\mathbb{T}'^i$ by $2(f + 2i - 2 - p)$ contractions and refinements.*

Proof. Observe that conversion of $\mathbb{T}^i$ to $\mathbb{T}'^i$ through minimum contractions and refinements requires all the internal edges, except those in the longest root-to-leaf path, to be contracted, and refined in the end. The total number of internal edges in $\mathbb{T}^i$ is $f + 2i - 2$, and the number of internal edges in the longest root-to-leaf path is p. Thus, $2(f + 2i - 2 - p)$ contraction and refinements must require to convert $\mathbb{T}^i$ to $\mathbb{T}'^i$. □

Lemma 28. *For i from 1 to n, all $\mathbb{T}^i$s can be converted to corresponding $\mathbb{T}'^i$s by $2(nf + n^2 - n - np)$ contractions and refinements.*

Proof. From Lemma 27,

$$\begin{aligned}\text{contractions and refinements} &= \sum_{i=1}^{n} 2(f + 2i - 2 - p) \\ &= 2(nf + n(n+1) - 2n - pn) \\ &= 2(nf + n^2 - n - np)\end{aligned}$$

□

Here is the connection between exactly covering S and transforming $\mathcal{T}_1$ into $\mathcal{T}_2$ by contractions and refinements: To transform $\mathcal{T}_1$ into $\mathcal{T}_2$, all we need is to convert each $\mathbb{T}^i$ on the left into $\mathbb{T}'^i$ and each tree $\mathbb{T}'^i$ on the right into $\mathbb{T}^i$. From Lemma 28, this costs $8(nf + n^2 - n - np)$ contractions and refinements, since for each $s_i \in S$, there is one $\mathbb{T}^i$ and three $\mathbb{T}'^i$s in $\mathcal{T}_1$. A rather clever technique is to swap all $\mathbb{T}^i$s on the left with their counterparts on the right and to manually transform the remaining $6q$ $\mathbb{T}'^i$s on the right into corresponding $\mathbb{T}^i$s.

Lemma 29. *For $C_i = \{s_{i_1}, s_{i_2}, s_{i_3}\}$ $(1 \leq i \leq n)$, all $\mathbb{T}'^{i_j}s$ $(1 \leq j \leq 3)$ can be swapped with corresponding $\mathbb{T}^{i_j}s$ by $2(2f + 1)$ contractions and refinements.*

Proof. Take the toll sequence corresponding for C_i and contract its $2f + 1$ edges; i.e., $(2f - 1)$ internal edges and 2 edges at both the sides of the toll sequence. Now refine it so that corresponding $\mathbb{T}^{i_j}$s move in C_i and $\mathbb{T}'^{i_j}$s stay in the left. This took $2(2f + 1)$ contractions and refinements. □

Theorem 14. *Set S has an exact cover if and only if the RF distance between $\mathcal{T}_1$ and $\mathcal{T}_2$ is not more than $\kappa = 2q(2f+1) + 4(nf + n^2 - n - np)$.*

Proof. Let an exact cover $C_{i_1}, ..., C_{i_q}$ of S exists. Now the $3q$ $\mathbb{T}^i$s can be partitioned into q groups according to the cover. For each $C_j = \{s_{j_1}, s_{j_2}, s_{j_3}\}$ (for $j = i_1, ..., i_q$) in the cover, we swap the corresponding $\mathbb{T}'$s (i.e., $\mathbb{T}'^{j_1}$, $\mathbb{T}'^{j_2}$, and $\mathbb{T}'^{j_3}$) with their counterparts (i.e., $\mathbb{T}^{j_1}$, $\mathbb{T}^{j_2}$, $\mathbb{T}^{j_3}$). From Lemma 29, this requires $2q(2f+1)$ contractions and refinements. There are 2 more copies of each $\mathbb{T}'^i$ $(1 \leq i \leq n)$ to convert into their corresponding $\mathbb{T}^i$. This requires $4(nf + n^2 - n - np)$ contractions and refinements (Lemma 28). Thus, if S has an exact cover, the RF distance between $\mathcal{T}_1$ and $\mathcal{T}_2$ is not more than $\kappa = 2q(2f+1) + 4(nf + n^2 - n - np)$.

For the other direction, let κ be divided into two parts: $\kappa_1 = 2q(2f+1)$ and $\kappa_2 = 4(nf + n^2 - n - np)$. Observe that $2q$ $\mathbb{T}'^i$s, in the right side of $\mathcal{T}_1$, always require to be transformed manually into their respective $\mathbb{T}^i$s. Thus, the κ_2 part of κ is fixed. We claim that if S has no exact cover than κ_1 is more than $2q(2f+1)$.

Let S has no exact cover. Now n $\mathbb{T}'^i$s on the left and n $\mathbb{T}'^i$s on the right side of $\mathcal{T}_1$ can be converted into their counterparts by three ways:

- **Swapping more than q triplets.** Let $q+\sigma$ triplets cover all elements in S (with some repeated elements). Now swapping n $\mathbb{T}^i$ with corresponding $\mathbb{T}'^i$ in $q+\sigma$ triplets will require $2(q+\sigma)(2f+1)$ contractions and refinements.

- **Swapping q triplets.** Let $C_{i_1}, ..., C_{i_q}$ be the best q triplets that cover all but σ elements in S. Swapping these q triplets only converts $n - \sigma$ $\mathbb{T}^i$s and $\mathbb{T}'^i$s into their counterparts. Rest 2σ $\mathbb{T}^i$s and $\mathbb{T}'^i$s need to be converted manually. Thus, κ_1 is $2q(2f+1) + <2\sigma$ manual conversions>.

- **Processing triplets.** Let $C_{i_1}, ..., C_{i_q}$ be the best q triplets that cover all but σ elements in S. First, we contract corresponding q toll sequences. We add or remove leaves in the rest σ $\mathbb{T}'^i$s (caterpillars) so that they correspond to the uncovered $\mathbb{T}^j$s on the left. Now $\mathbb{T}'^i$ moved to left and $\mathbb{T}^i$ to right. Before refining the cental vertex, the $\mathbb{T}^i$s corresponding to processed $\mathbb{T}'^i$s are reprocessed to correspond for the original caterpillar mul-tree. This adds processing cost for uncovered σ elements of S in κ_1 in addition to $2q(2f+1)$.

All the above three ways of converting $\mathcal{T}_1$ into $\mathcal{T}_2$, have κ_1 that is more than $2q(2f+1)$. $\square$

Thus, computing the RF distance between two uniform, mul-trees is NP-complete, and together with Theorem 13, we complete the NP-completeness proof of the tree labeling problem.

CHAPTER 8. Conclusion

Phylogenies are of central importance to biology, and so is the construction of the tree of all life on earth. The proliferation of next generation sequencing technologies has presented extraordinary opportunities, but it also has drawn attention to many complex computational problems. This thesis addressed several of theoretical, computational was well as experimental, computational biology problems that arise along the way towards building the tree of life.

First, we developed a heuristic method for NP-hard unrooted Robinson-Foulds (RF) supertree problem, and showed that it yields more accurate supertrees than those obtained from Matrix Representation with Parsimony (MRP) and the rooted RF heuristic. For the future, it appears to be important to incorporate uncertainty within the input trees into an RF supertree analysis by weighting the splits when calculating the RF distance.

Inferring species trees from conflicting multi-copy gene trees is a critical problem in phylogenetics. Most previous methods assume that the gene tree conflict is caused by a specific biological process such as gene duplication and loss, deep coalescence, or lateral gene gene transfer. We presented an RF distance measure based approach (MulRF) to infer a species tree from input multi-copy gene trees, through a generalization of RF distance to multi-labeled trees. Simulation experiments have shown that this approach produces more accurate species trees than existing methods when incongruence is caused by gene tree error, duplications and losses, and/or lateral gene transfer. The effectiveness of the MulRF method suggests that other tree distance metrics (such as quartet distance) can also be used in inferring species trees from multi-copy gene trees, opening the doors for further research.

Perhaps the most frustrating aspect of phylogenetics is the myriad of available species tree inference methods, and the lack of any formal comparative study on the performance of some of these methods. Our simulation study fills this void, providing a thorough evaluation of the

performance of Gene Tree Parsimony (GTP) under duplication and duplication and loss cost models and a comparison with our MulRF method. We particularly looked at the effects of various samplings (e.g., gene tree and sequence sampling), gene tree error, and duplication and loss rates on the accuracy of the phylogenetic estimates by GTP and MulRF. Our results highlighted the difficulty in inferring species trees accurately for decreased gene tree and sequence sampling, and increased duplication and loss rates. In general, MulRF was best in estimating small species trees ($\leq$ 100 taxa), and duplication and loss cost based GTP for larger species trees ($\geq$ 250 taxa).

We presented efficient error correction algorithms for gene tree reconciliation based on duplication, duplication and loss, and deep coalescence. In particular, these algorithms rapidly search local Subtree Prune and Regraft (SPR) or Tree Bisection and Reconnection (TBR) neighborhoods of a given gene tree to identify a topology that implies the fewest of these evolutionary events for a given species tree. One simple extension of this work may be to weight the possible gene tree rearrangements based on support for different clades in the gene tree. Thus, well-supported clades may be rarely or never be subject to rearrangement, while poorly supported clades may be subject to extensive rearrangements.

Finally, we also presented NP-completeness proofs for two open problems in phylogenetics. The first problem is computing the RF distance between two multi-labeled trees. The second problem is a tree labeling problem: Labeling two unlabeled trees so as to minimize the RF distance between the resulting singly-labeled trees. These results help redirecting the future research towards designing approximation algorithm for these problems, rather than searching endlessly for exact solutions.

My thesis not only makes useful theoretical additions in the computational biology literature, but also provides implementations for most of our methods for further research. Further, our research also draws attention to many, new problems. I believe that exploring them will be fruitful for the phylogenetics research community.

APPENDIX A. Commonly used symbols

Chapter 3,4,7

T	Unrooted phylogenetic tree
$\mathcal{L}(T)$	Leaf set of tree T
$V(T)$	Set of all vertices of T
$E(T)$	Set of all edges of T
$I(T)$	Set of internal vertices of T
$T_{\|U}$	Restriction of T to U
$\Sigma(T)$	Set of all non-trivial splits of T
$\mathbb{T}$	Rooted phylogenetic tree
$rt(\mathbb{T})$	Root of tree $\mathbb{T}$
$\mathbb{T}_v$	Subtree of T rooted at v
$\mathcal{H}(\mathbb{T})$	Set of all clusters of $\mathbb{T}$
$\mathcal{T}$	Unrooted mul-tree

Chapter 6

T	Rooted phylogenetic tree
$\mathcal{L}(T)$	Leaf set of tree T
$V(T)$	Set of all vertices of T
$E(T)$	Set of all edges of T
$I(T)$	Set of internal vertices of T
$T_{\|U}$	Restriction of T to U
$Pa_T(v)$	Parent of vertex v in T

(u, v)	Edge u, v, if u is parent of v
$Ch_T(v)$	Set of children of v in T
$Sb_T(w)$	Sibling vertex of w in T
$d_T(x, y)$	Number of edges on the unique path between x and y in T

BIBLIOGRAPHY

Allen, B. L. and Steel, M. (2001). Subtree transfer operations and their induced metrics on evolutionary trees. *Annals of Combinatorics*, 5:1–15.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215:403—410.

Ané, C., Larget, B., Baum, D. A., Smith, S. D., and Rokas, A. (2007). Bayesian Estimation of Concordance Among Gene Trees. *Mol. Biol. Evol.*, 24(7):1575.

Arvestad, L., Berglund, A.-C., Lagergren, J., and Sennblad, B. (2003). Bayesian gene/species tree reconciliation and orthology analysis using MCMC. *Bioinformatics*, 19(Suppl 1):i7–i15.

Arvestad, L., Berglund, A.-C., Lagergren, J., and Sennblad, B. (2004). Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. In *RECOMB*, pages 326–335.

Arvestad, L., Lagergren, J., and Sennblad, B. (2009). The gene evolution model and computing its associated probabilities. *Journal of the ACM*, 56(2):1–44.

Bansal, M. S., Burleigh, J. G., and Eulenstein, O. (2010a). Efficient genome-scale phylogenetic analysis under the duplication-loss and deep coalescence cost models. *BMC Bioinformatics*, 11(Suppl 1):S42.

Bansal, M. S., Burleigh, J. G., Eulenstein, O., and Fernández-Baca, D. (2010b). Robinson-Foulds supertrees. *Algorithms for Molecular Biology*, 5:18.

Baum, B. R. (1992). Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. *Taxon*, 41:3–10.

Beck, R. M. D., Bininda-Emonds, O. R. P., Cardillo, M., Liu, F. R., and Purvis, A. (2006). A higher-level MRP supertree of placental mammals. *BMC Evolutionary Biology*, 6:93.

Beiko, R. G., Doolittle, W. F., and Charlebois, R. L. (2008). The impact of reticulate evolution on genome phylogeny. *Systematic Biology*, 57:844–856.

Bender, M. A. and Farach-Colton, M. (2000). The LCA problem revisited. In Gonnet, G. H., Panario, D., and Viola, A., editors, *LATIN*, volume 1776 of *Lecture Notes in Computer Science*, pages 88–94. Springer.

Berglund-Sonnhammer, A., Steffansson, P., Betts, M. J., and Liberles, D. A. (2006). Optimal gene trees from sequences and species trees using a soft interpretation of parsimony. *Journal of Molecular Evolution*, 63:240–250.

Bininda-Emonds, O. R. P., Beck, R. M. D., and Purvis, A. (2005). Getting to the roots of matrix representation. *Syst. Biol.*, 54:668–672.

Bininda-Emonds, O. R. P., Cardillo, M., Jones, K. E., MacPhee, R. D. E., Beck, R. M. D., Grenyer, R., Price, S. A., Vos, R. A., Gittleman, J. L., and Purvis, A. (2007). The delayed rise of present-day mammals. *Nature*, 446:507–512.

Bininda-Emonds, O. R. P. and Sanderson, M. J. (2001). Assessment of the accuracy of matrix representation with parsimony analysis supertree construction. *Systematic Biology*, 50:565–579.

Bordewich, M. and Semple, C. (2004). On the computational complexity of the rooted subtree prune and regraft distance. *Annals of Combinatorics*, 8:409–423.

Burleigh, J. G., Bansal, M. S., Eulenstein, O., Hartmann, S., Wehe, A., and Vision, T. J. (2011). Genome-scale phylogenetics: Inferring the plant tree of life from 18,896 gene trees. *Systematic Biology*, 60(2):117–125.

Burleigh, J. G., Bansal, M. S., Eulenstein, O., and Vision, T. J. (2010). Inferring species trees from gene duplication episodes. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 198–203.

Burleigh, J. G., Bansal, M. S., Wehe, A., and Eulenstein, O. (2009). Locating large-scale gene duplication events through reconciled trees: Implications for identifying ancient polyploidy events in plants. *Journal of Computational Biology*, 16:1071–1083.

Bush, R. M., Bender, C. A., Subbarao, K., Cox, N. J., and Fitch, W. M. (1999). Predicting the evolution of human influenza a. *Science*, 286:1921–1925.

Cardillo, M., Bininda-Emonds, O. R. P., Boakes, E., and Purvis, A. (2004). A species-level phylogenetic supertree of marsupials. *Journal of Zoology*, 264:11–31.

Chang, W., Burleigh, J. G., Fernández-Baca, D., and Eulenstein, O. (2011). An ILP solution for the gene duplication problem. *BMC Bioinformatics*, 12(Suppl 1):S14.

Chaudhary, R., Burleigh, J. G., and Eulenstein, O. (2012a). Efficient error correction algorithms for gene tree reconciliation based on duplication, duplication and loss, and deep coalescence. *BMC Bioinformatics*, 13:S11.

Chaudhary, R., Burleigh, J. G., and Fernández-Baca, D. (2012b). Fast local search for unrooted Robinson-Foulds supertrees. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9:1004–1013.

Chaudhary, R., Burleigh, J. G., and Fernández-Baca, D. (2012c). Inferring species trees from incongruent multi-copy gene trees using the Robinson-Foulds distance. (under review).

Chen, D., Eulenstein, O., Fernández-Baca, D., and Burleigh, J. G. (2006). Improved heuristics for minimum-flip supertree construction. *Evolutionary Bioinformatics*, 2:347–356.

Chen, K., Durand, D., and Farach-Colton, M. (2000). Notung: a program for dating gene duplications and optimizing gene family trees. *Journal of Computational Biology*, 7:429–447.

Cotton, J. A. and Page, R. D. M. (2002). Going nuclear: gene family evolution and vertebrate phylogeny reconciled. *P. Roy. Soc. Lond. B Biol.*, 269:1555–1561.

Cotton, J. A. and Page, R. D. M. (2003). Gene tree parsimony vs. uninode coding for phylogenetic reconstruction. *Molecular Phylogenetics and Evolution*, 29:298–308.

Creevey, C. J. and McInerney, J. O. (2005). Clann: Investigating phylogenetic information through supertree analyses. *Bioinformatics*, 21(3):390–392.

Davies, T. J., Barraclough, T. G., Chase, M. W., Soltis, P. S., Soltis, D. E., and Savolainen, V. (2004). Darwin's abominable mystery: insights from a supertree of the angiosperms. *Proceedings of the National Academy of Sciences of the United States of America*, 101:1904–1909.

Durand, D., Halldórsson, B. V., and Vernot, B. (2006). A hybrid micro-macroevolutionary approach to gene tree reconstruction. *Journal of Computational Biology.*, 13(2):320–335.

Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32:1792—1797.

Erwin, T. L. (1991). An evolutionary basis for conservation strategies. *Science*, 253:750–752.

Eulenstein, O. (1998). *Predictions of gene-duplications and their phylogenetic development.* PhD thesis, University of Bonn, Germany. GMD Research Series No. 20 / 1998, ISSN: 1435-2699.

Eulenstein, O., Chen, D., Burleigh, J. G., Fernández-Baca, D., and Sanderson, M. J. (2004). Performance of flip supertree construction with a heuristic algorithm. *Systematic Biology*, 53:299–308.

Feller, W. (1968). *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition.* Wiley, 3 edition.

Felsenstein, J. (1993). Retree software. http://evolution.genetics.washington.edu/phylip/doc/retree.html.

Ganapathy, G. (2006). *Algorithms and Heuristics for Combinatorial Optimization in Phylogeny.* PhD thesis, University of Taxas at Austin.

Ganapathy, G., Goodson, B., Jansen, R., Le, H., Ramachandran, V., and Warnow, T. (2006). Pattern identification in biogeography. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3:334–346.

Ganapathy, G., Ramachandran, V., and Warnow, T. (2003). Better hill-climbing searches for parsimony. In Benson, G. and Page, R. D. M., editors, *WABI*, volume 2812 of *Lecture Notes in Computer Science*, pages 245–258. Springer.

Ganapathy, G., Ramachandran, V., and Warnow, T. (2004). On contract-and-refine transformations between phylogenetic trees. In Munro, J. I., editor, *SODA*, pages 900–909. SIAM.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A guide to the theory of NP-completeness*. W. H. Freeman, New York.

Ge, F., Wang, L.-S., and Kim, J. (2005). The cobweb of life revealed by genome-scale estimates of horizontal gene transfer. *PLoS Biology*, 3:909–914.

Goloboff, P. A. (1999). Analyzing large data sets in reasonable times: Solutions for composite optima. *Cladistics*, 15:415–428.

Goloboff, P. A. (2005). Minority rule supertrees? MRP, compatibility, and minimum flip may display the least frequent groups. *Cladistics*, 21:282–294.

Goodman, M., Czelusniak, J., Moore, G. W., Romero-Herrera, A. E., and Matsuda, G. (1979). Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Biology*, 28:132–163.

Górecki, P., Burleigh, J. G., and Eulenstein, O. (2011). Maximum likelihood models and algorithms for gene tree evolution with duplications and losses. *BMC Bioinformatics*, 12:S15.

Górecki, P., Burleigh, J. G., and Eulenstein, O. (2012). GTP supertrees from unrooted gene trees: Linear time algorithms for NNI based local searches. In *ISBRA*, pages 102–114.

Górecki, P. and Tiuryn, J. (2006). Inferring phylogeny from whole genomes. In *ECCB (Supplement of Bioinformatics)*, pages 116–122.

Guigó, R., Muchnik, I., and Smith, T. F. (1996). Reconstruction of ancient molecular phylogeny. *Molecular Phylogenetics and Evolution*, 6(2):189–213.

Hahn, M. W. (2007). Bias in phylogenetic tree reconciliation methods: implications for vertebrate genome evolution. *Genome Biology*, 8:R141.

Heled, J. and Drummond, A. J. (2010). Bayesian inference of species trees from multilocus data. *Journal of Molecular Biology and Evolution*, 27:570–580.

Hickey, G., Dehne, F., Rau-Chaplin, A., and Blouin, C. (2008). SPR distance computation for unrooted trees. *Evolutionary Bioinformatics*, 4:17–27.

Holland, B., Penny, D., and Hendy, M. (2003). Outgroup misplacement and phylogenetic inaccuracy under a molecular clock —- a simulation study. *Syst. Biol.*, 52:229–238.

Holton, T. A. and Pisani, D. (2010). Deep genomic-scale analyses of the metazoa reject coelomata: Evidence from single- and multigene families analyzed under a supertree and supermatrix paradigm. *Genome Biology and Evolution*, 2:310—324.

Huang, H. and Knowles, L. L. (2009). What is the danger of the anomaly zone for empirical phylogenetics? *Systematic Biology*, 58:527–536.

Huelsenbeck, J. P., Bollback, J. P., and Levine, A. M. (2002). Inferring the root of a phylogenetic tree. *Syst. Biol.*, 51:32–43.

Joly, S. and Bruneau, A. (2009). Measuring branch support in species trees obtained by gene tree parsimony. *Systematic Biology*, 58:100–113.

Katz, L. A., Grant, J. R., Parfrey, L. W., and Burleigh, J. G. (2012). Turning the crown upside down: Gene tree parsimony roots the eukaryotic tree of life. *Systematic Biology*, 61:653–660.

Kubatko, L. S., Carstens, B. C., and Knowles, L. L. (2009). STEM: species tree estimation using maximum likelihood for gene trees under coalescence. *Bioinformatics*, 25(7):971–973.

Kubatko, L. S. and Degnan, J. H. (2007). Inconsistency of phylogenetic estimates from concatenated data under coalescence. *Systematic Biology*, 56(1):17–24.

Leebens-Mack, J., Raubeson, L. A., Cui, L., Kuehl, J. V., Fourcade, M. H., Chumley, T. W., Boore, J. L., Jansen, R. K., and dePamphilis, C. W. (2005). Identifying the basal angiosperm

node in chloroplast genome phylogenies: sampling one's way out of the Felsenstein zone. *Mol. Biol. Evol.*, 22:1948–1963.

Liu, L. (2008). BEST: Bayesian estimation of species trees under the coalescent model. *Bioinformatics*, 24(21):2542–2543.

Liu, L. and Pearl, D. K. (2007). Species Trees from Gene Trees: Reconstructing Bayesian Posterior Distributions of a Species Phylogeny Using Estimated Gene Tree Distributions. *Systematic Biology*, 56(3):504–514.

Liu, L., Yu, L., Kubatko, L., Pearl, D. K., and Edwards, S. V. (2009). Coalescent methods for estimating phylogenetic trees. *Molecular Phylogenetics and Evolution*, 53:320–328.

Lloyd, G. T., Davis, K. E., Pisani, D., Tarver, J. E., Ruta, M., Sakamoto, M., Hone, D. W. E., Jennings, R., and Benton, M. J. (2008). Dinosaurs and the cretaceous terrestrial revolution. *Proc. Roy. Soc. B*, 275:2483–2490.

Lomolino, M. V., Riddle, B. R., Whittaker, R. J., and Brown, J. H. (2005). *Biogeography*. Sinauer Associates, Sunderlan, MA, Chicago, 3 edition.

Maddison, W. P. (1997). Gene trees in species trees. *Systematic Biology*, 46:523–536.

Maddison, W. P. and Maddison, D. (2009). Mesquite: a modular system for evolutionary analysis. version 2.6. http://mesquiteproject.org.

McMorris, F. R. and Steel, M. A. (1993). The complexity of the median procedure for binary trees. In *In Proceedings of the International Federation of Classification Societies*.

Medina, E. M., Jones, G. W., and Fitzpatrick, D. A. (2011). Reconstructing the fungal tree of life using phylogenomics and a preliminary investigation of the distribution of yeast prion-like proteins in the fungal kingdom. *Journal of Molecular Evolution*, 73:116–133.

Mossel, E. and Vigoda, E. (2005). Phylogenetic MCMC algorithms are misleading on mixtures of trees. *Science*, 309(5744):2207–2209.

Near, T. J., Eytan, R. I., Dornburg, A., Kuhn, K. L., Moore, J. A., Davis, M. P., Wainwright, P. C., Friedman, M., and Smith, W. L. (2012). Resolution of ray-finned fish phylogeny and timing of diversification. *Proceedings of the National Academy of Sciences*, 109:13698–13703.

Ness, R. W., Graham, S. W., and Barrett, S. C. H. (2011). Reconciling gene and genome duplication events: Using multiple nuclear gene families to infer the phylogeny of the aquatic plant family pontederiaceae. *Journal of Molecular Biology and Evolution*, 28:3009–3018.

Nixon, K. C. (1999). The parsimony ratchet: a new method for rapid parsimony analysis. *Cladistics*, 15:407–414.

Page, R. D. M. (1988). Quantitative cladistic biogeography: Constructing and comparing area cladograms. *Systematic Zoology*, 37(1):254–270.

Page, R. D. M. (1994). Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. *Systematic Biology*, 43(1):58–77.

Page, R. D. M. (1998). GeneTree: comparing gene and species phylogenies using reconciled trees. *Bioinformatics*, 14(9):819–820.

Penny, D., White, W. T., Hendy, M. D., and Phillips, M. J. (2008). A bias in ML estimates of branch lengths in the presence of multiple signals. *Molecular Biology and Evolution*, 25(2):239–242.

Pisani, D. and Wilkinson, M. (2002). Matrix representation with parsimony, taxonomic congruence and total evidence. *Systematic Biology*, 51:151–155.

Pisani, D., Yates, A. M., Langer, M. C., and Benton, M. J. (2002). A genus-level supertree of the Dinosauria. *Proceedings of the Royal Society of London*, 269:915–921.

Purvis, A. (1995). A modification to Baum and Ragan's method for combining phylogenetic trees. *Systematic Biology*, 44:251–255.

Ragan, M. A. (1992). Phylogenetic inference based on matrix representation of trees. *Molecular Phylogenetics and Evolution*, 1:53–58.

Åkerborg, O., Sennblad, B., Arvestad, L., and Lagergren, J. (2009). Simultaneous Bayesian gene tree reconstruction and reconciliation analysis. *Proceedings of the National Academy of Sciences*, 106(14):5714–5719.

Rambaut, A. and Grassly, N. C. (1997). Seq-Gen: An application for the Monte-Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, 13:235–238.

Rasmussen, M. D. and Kellis, M. (2011). A bayesian approach for fast and accurate gene tree reconstruction. *Journal of Molecular Biology and Evolution*, 28:273–290.

Rasmussen, M. D. and Kellis, M. (2012). Unified modeling of gene duplication, loss, and coalescence using a locus tree. *Genome Research*, 22:755—765.

Robinson, D. F. and Foulds, L. R. (1981). Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147.

Rokas, A., Williams, B. L., King, N., and Carroll, S. B. (2003). Genome-scale approaches to resolving incongruence in molecular phylogenies. *Nature*, 425(6960):798–804.

Sanderson, M. J. (2003). r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. *Bioinformatics*, 19:301–302.

Sanderson, M. J. and McMahon, M. M. (2007a). Inferring angiosperm phylogeny from EST data with widespread gene duplication. *BMC Evolutionary Biology*, 7(suppl 1:S3).

Sanderson, M. J. and McMahon, M. M. (2007b). Inferring angiosperm phylogeny from EST data with widespread gene duplication. *BMC Evolutionary Biology*, 7(Suppl 1):S3.

Sanderson, M. J. and Shaffer, H. B. (2002). Troubleshooting molecular phylogenetic analyses. *Annu. Rev. Ecol. Evol. Syst.*, 33:49–72.

Semple, C. and Steel, M. (2003). *Phylogenetics*. Oxford University Press.

Slowinski, J. B., Knight, A., and Rooney, A. P. (1997). Inferring species trees from gene trees: A phylogenetic analysis of the elapidae (serpentes) based on the amino acid sequences of venom proteins. *Molecular Phylogenetics and Evolution*, 8:349–362.

Smith, A. B. (1994). Rooting molecular trees: problems and strategies. *Biol. J. Linn. Soc.*, 51:279–292.

Stamatakis, A. (2006a). RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22:2688—2690.

Stamatakis, A. (2006b). RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688–2690.

Steel, M. and Rodrigo, A. (2008). Maximum likelihood supertrees. *Systematic Biology*, 57(2).

Swenson, M. S., Barbançon, F., Warnow, T., and Linder, C. R. (2010). A simulation study comparing supertree and combined analysis methods using SMIDGen. *Algorithms for Molecular Biology*, 5:8.

Swenson, M. S., Suri, R., Linder, C. R., and Warnow, T. (2011). An experimental study of quartets maxcut and other supertree methods. *Algorithms for Molecular Biology*, 6:7.

Swofford, D. L. (2003). PAUP*: Phylogenetic analysis using parsimony (*and other methods), version 4.0.

Than, C. and Nakhleh, L. (2009). Species tree inference by minimizing deep coalescences. *PLoS Comput Biol*, 5(9):e1000501.

Than, C. V. and Rosenberg, N. A. (2011). Consistency properties of species tree inference by minimizing deep coalescences. *Journal of Computational Biology*, 18:1–15.

Vernot, B., Stolzer, M., Goldman, A., and Durand, D. (2007). Reconciliation with non-binary species trees. *Computational Systems Bioinformatics*, 53:441–452.

Wainwright, P. C., Smith, W. L., Price, S. A., Tang, K. L., Sparks, J. S., Ferry, L. A., Kuhn, K. L., Eytan, R. I., and Near, T. J. (2012). The evolution of pharyngognathy: A phylogenetic and functional appraisal of the pharyngeal jaw key innovation in labroid fishes and beyond. *Systematic Biology*, 0:1–27.

Wehe, A., Bansal, M. S., Burleigh, J. G., and Eulenstein, O. (2008). Duptree: a program for large-scale phylogenetic analyses using gene tree parsimony. *Bioinformatics*, 24(13):1540–1541.

Wehe, A. and Burleigh, J. G. (2010). Scaling the gene duplication problem towards the tree of life. In Al-Mubaid, H., editor, *BICoB*, pages 133–138. ISCA.

Wheeler, W. C. (1990). Nucleic acid sequence phylogeny and random outgroups. *Cladistics*, 6:363–367.

Whidden, C., Zeh, N., and Beiko, R. (2012). SPRSupertrees. version 1.1.0. http://kiwi.cs.dal.ca/software/sprsupertrees.

Yap, V. B. and Speed, T. (2005). Rooting a phylogenetic tree with nonreversible substitution models. *BMC Evol. Biol.*, 5:2.

Yu, Y., Warnow, T., and Nakhleh, L. (2011). Algorithms for MDC-based multi-locus phylogeny inference. In *RECOMB*, pages 531–545.

Zhang, L. (1997). On a Mirkin-Muchnik-Smith conjecture for comparing molecular phylogenies. *Journal of Computational Biology*, 4(2):177–187.