ALGORITHMS FOR PHYLOGENETIC TREE CORRECTION IN SPECIES AND CANCER EVOLUTION

BY

SARAH CHRISTENSEN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Doctoral Committee:

Assistant Professor Mohammed El-Kebir, Chair and Director of Research
Professor Tandy Warnow, Co-Director of Research
Professor Sariel Har-Peled
Professor Luay Nakhleh, Rice University

ABSTRACT

Reconstructing evolutionary trees, also known as *phylogenies*, from molecular sequence data is a fundamental problem in computational biology. Classically, evolutionary trees have been estimated over a set of species, where leaves correspond to extant species and internal nodes correspond to ancestral species. This type of phylogeny is colloquially thought of as the "Tree of Life" and assembling it has been designated as a Grand Challenge by the National Science Foundation Advisory Committee for Cyberinfrastructure. However, processes other than speciation are also shaped by evolution. One notable example is in the development of a malignant tumor; tumor cells rapidly grow and divide, acquiring new mutations with each subsequent generation. Tumor cells then compete for resources, often resulting in selection for more aggressive cell types. Recent advancements in sequencing technology rapidly increased the amount of sequencing data taken from tumor biopsies. This development has allowed researchers to attempt reconstructing evolutionary histories for individual patient tumors, improving our understanding of cancer and laying the groundwork for precision therapy.

Despite algorithmic improvements in the estimation of both species and tumor phylogenies from molecular sequence data, current approaches still suffer a number of limitations. Incomplete sampling and estimation error can lead to missing leaves and low-support branches in the estimated phylogenies. Moreover, commonly posed optimization problems are often under-determined given the limited amounts and low quality of input data, leading to large solution spaces of equally plausible phylogenies. In this dissertation, we explore current limitations in both species and tumor phylogeny estimation, connecting similarities and highlighting key differences. We then put forward four new methods that improve phylogeny estimation methods by incorporating auxiliary information: OCTAL, TRACTION, PhySigs, and RECAP. For each method, we present theoretical results (e.g., optimization problem complexity, algorithmic correctness, running time analysis) as well as empirical results on simulated and real datasets. Collectively, these methods show we can significantly improve the accuracy of leading phylogeny estimation methods by leveraging additional signal in distinct, but related datasets.

ACKNOWLEDGEMENTS

I would like to begin by thanking my advisors, without whom this dissertation would not be possible. In particular, I would like to thank Professor Tandy Warnow for taking a chance on me when I first started my Ph.D. with no background in computer science. I am especially grateful for her candor when giving advice, her patience as I navigated my way through graduate school, and her support as I went on to explore new topics. She is truly a role model to young women scientists, and I benefited immensely from her mentorship. I would also like to thank Professor Mohammed El-Kebir whose cheerful demeanor and optimism helped me to keep pushing through the ups and downs of graduate school. It is through Mohammed that I was introduced to cancer genomics, which connected the two types of evolution that ultimately became my dissertation. Most of all, I appreciate Mohammed for always encouraging me follow my own interests, even when it led to him sitting through long talks on complexity lower bounds.

I also want to acknowledge other important people that helped my academic progression. Specifically, I would like to thank my Doctoral Committee, including Professor Sariel Har-Peled and Professor Luay Nakhleh, for their thoughtful feedback and guidance. Professor Chandra Chekuri was also extremely generous with his time throughout my Ph.D. experience, often offering a helpful outside perspective. Moreover, I would be remiss if I did not thank all of my coauthors: Juho Kim, Erin K. Molloy, Pranjal Vachaspati, Ananya Yammanuru, Professor Nicholas Chia, Professor Oluwasanmi Koyejo, and Professor Max Leiserson. It has been immensely rewarding to collaborate with each of these individuals. I am additionally grateful to the University of Illinois at Urbana-Champaign staff, especially Candice Steidinger, Viveka Kudaligama, Kara MacGregor, and Maggie Metzger Chappell, for keeping me on track to graduate and for keeping the school functioning in the pandemic.

I would next like to acknowledge all of the support I received from my fellow graduate students, without whom I would not have made it past my first semester, let alone my dissertation. Particularly, I would like to thank Ehsan Saleh for helping me not fail the first class of my Ph.D. program. Despite going through his own challenges, he still found so much time to help me, and I will never forget that kindness. I would also like to thank Allyson Lauren Kaminsky for being a great roommate and an even better friend. Her own personal growth throughout graduate school has been an inspiration to me, and I always value her perspective. Next, I would like to thank Erin K! Molloy and her sweet cat Alice; I am so lucky to have been able to (try to) follow in the footsteps of such a successful researcher and

learn from her example. Lastly, I would like to acknowledge the Warnow and El-Kebir lab groups, who fostered an intellectual community that helped me grow.

Of course, I am also grateful for the financial support of the Chirag Fellowship (2016-2018), State Farm Doctoral Award (Spring 2018), C.L. and Jane Liu Award (Spring 2018), Ira & Debra Cohen Graduate Fellowship (Spring 2019), and the National Science Foundation (through Grant No. CCF-1535977 and Grant No. IIS 15-13629 to Tandy Warnow). Many computational experiments presented in this dissertation were performed on the Illinois Campus Cluster. The Illinois Campus Cluster is a computing resource supported by funds from the University of Illinois at Urbana-Champaign and is operated by the Illinois Campus Cluster Program in conjunction with the National Center for Supercomputing Applications.

Finally, I would like to thank my close friends and family for their love and encouragement as I pursued this final stage of my formal education. I am grateful to my mother, who sacrificed so much to focus on raising me. She likes to tell the story of how when I was little I asked her for the first time about the concept of addition on a day where she was sick in bed. Not wanting to pass up on a teaching moment, she got up and proceeded to encourage me for the rest of the afternoon, teaching me addition and subtraction. While I have heard this story many times in the context of my mother boasting about me, I repeat it here because I think it also shows much about her; she is such a strong woman who has always pushed me to take on new challenges. I am likewise thankful for my father whose hard work and dedication provided me with so much opportunity. He has always been my number one fan, once driving over four hours to watch me in an ice skating competition that only lasted 3 minutes, only to turn around and drive back for work. He instilled in me the work ethic and diligence that allowed me to persevere through this program. I would next like to thank both of my grandmothers who were in many ways ahead of their time. They have shown me that learning is a life long activity and that one can have intellectual curiosity at any age. Last but not least, I want to thank Kent Quanrud for being my best friend and champion throughout this experience. Kent helped me dream big and feel as if, with computers and imagination, anything is possible.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

The theory of evolution, which states that changes in heritable characteristics of organisms are shaped by random mutation and natural selection over the course of generations, is on the surface a simple and elegant idea. Yet, this idea has grown to have far-reaching implications underpinning much of biology. Here, we discuss computational methods aimed at reconstructing evolutionary history in two different contexts: species and tumor evolution.

The initial conception of evolutionary theory was famously an attempt to explain the great variety of life on Earth. Contemporary understandings of evolution have, of course, moved beyond Darwinian theory to incorporate modern genetics. We now know that heritable characteristics are passed down through *genes*, where the complete set of genes for an organism is collectively known as a *genome*. Genomes are encoded in special molecules, called DNA, which are chains comprised of four varieties of nucleotides (Adenine, Cytosine, Guanine, and Thymine). Modern genetic sequencing technology allows us to reconstruct the genomes of different organisms, which we denote as a string of characters over an alphabet of four letters (A, C, G, and T, respectively). We now can reconstruct evolutionary relationships between species not with macro-level traits, but with this micro-level genetic material under specialized models of sequence evolution. We typically represent the resulting evolutionary relationships between species with graphical models, known formally as *species phylogenies* and popularly as *evolutionary trees*. There are two fundamental subtypes of phylogenies in this context: *species trees*, which describe the evolutionary history for populations of organisms and are estimated from multi-locus datasets, and *gene trees*, which describe the evolutionary history of a gene and are estimated from just a single locus. Large-scale sequencing projects generating the necessary inputs for species phylogenies are currently underway, including the 5000 Insect Genomes Project [1, 2], the 10000 Plant Genomes Project [3], and the Earth BioGenome Project [4], taking us one step closer to the ultimate vision of integrating life on Earth into a unified Tree of Life.

More recently, the principles of evolutionary theory have also been applied to tumorigenesis, the progression of tumors [5]. A malignant tumor is characterized by a fast proliferation of cells that accumulate new *somatic mutations* with each subsequent generation. *Intra-tumor heterogeneity* is in fact the primary cause of relapse and resistance to treatment, a major contributing factor for why cancer remains a leading cause of premature death globally [6, 7, 8]. We use *tumor phylogenies* to study this heterogeneity by reconstructing evolutionary relationships between tumor *clones*, groups of cells with nearly identical sets of mutations. Large-scale sequencing projects to facilitate our understanding of tumor composition are likewise underway. For instance, The Cancer Genome Atlas (TCGA) has already sequenced over 20,000 tumors spanning 33 cancer types and anticipates generating 2.5 petabytes of genomic data [9]. It is the hope that sufficient tumor sequencing coupled with tailored algorithms will help uncover evolutionary mechanisms driving tumorigenesis, enabling clinically-relevant tumor subtyping and ultimately improving patient care.

Much progress has already been made on developing computational methods for estimating phylogenies in both contexts. As with anything in science, there are still a number of limitations. While many of these limitations are shared, some stem from context specific details that do not translate between species and tumor phylogenies. For example, biological processes such as gene duplication and loss (GDL), incomplete lineage sorting (ILS), and horizontal gene transfer (HGT) create heterogeneous gene and species tree topologies, complicating the inference process [10]. We discuss each context below separately to carefully handle these important distinctions.

This dissertation is therefore conceptually divided into two parts. The first part (Chapters 3-4) begins with the formulation of optimization problems and algorithms in the context of species phylogeny estimation. The second part (Chapters 5-6) then uses a similar framework and toolkit to pose and solve problems in the context of tumor evolution. In both cases, we address the limitations of current methods by leveraging auxiliary information, such as sequencing data from other genes or other patients, to *correct* low-quality phylogenies output by existing methods or *prioritize* phylogenies when there are multiple optimal solutions. Along the way, many of our methods give additional insight into the broader mechanics of evolution beyond improving individual phylogenies.

Chapter 3 starts by addressing the challenge of correcting estimated gene trees that are missing species. To do so, we rely on a *reference tree*, typically estimated from regions of the genome outside of the missing gene. We formalize our objective by introducing the *Optimal Tree Completion* (OTC) problem, a general optimization problem that involves adding missing leaves to an unrooted binary tree so as to minimize its distance to a reference tree on a superset of the leaves. In particular, given a pair (T, t) of unrooted binary trees on leaf sets $(S,\ R \subseteq S)$, we wish to add all leaves from $S \setminus R$ to t in such a way that minimizes the distance to T. We show that when the distance is defined by one of the most common measures in phylogenetics, the *Robinson-Foulds* (RF) distance [11], an optimal solution can be found in polynomial time. We do this constructively by presenting OCTAL, an algorithm that solves the RF-OTC problem exactly in $O(|S|^2)$ time. We then report on a simulation study where we complete estimated gene trees using a reference tree estimated from a multilocus dataset. OCTAL produces completed gene trees that are closer to the true gene trees than the only previously existing heuristic, but the accuracy of the completed gene trees computed by OCTAL depends on how topologically similar the estimated species tree is to the true gene tree. Hence, under conditions with relatively low gene tree heterogeneity, OCTAL can be used to provide highly accurate completions.

Chapter 4 expands on the framework of OCTAL in order to also correct weakly supported branches in estimated gene trees using a reference tree. While previous work only focused on performing gene tree branch correction in the context of GDL, here we address gene tree correction where heterogeneity is instead due to ILS (a common problem in eukaryotic phylogenetics) and HGT (a common problem in bacterial phylogenetics). We introduce TRACTION, a simple polynomial time method that provably finds an optimal solution to the *Robinson-Foulds Optimal Tree Refinement and Completion* (RF-OTRC) problem. This problem we formulate seeks a refinement and completion of an input tree t with respect to a given binary reference tree T so as to minimize the RF distance. In practice, t is typically an estimated gene tree whose low-support edges have been collapsed. We present the results of an extensive simulation study evaluating TRACTION within gene tree correction pipelines on 68,000 estimated gene trees, using reference trees estimated from multi-locus data. We explore accuracy under conditions with varying levels of gene tree heterogeneity due to ILS and HGT. We show that TRACTION matches or improves the accuracy of well-established methods from the GDL literature under conditions with HGT and ILS, and ties for best under the ILS-only conditions. Furthermore, TRACTION ties for fastest on these datasets.

Chapter 5 then pivots into the context of tumor phylogeny estimation. Here we propose an approach that prioritizes alternative phylogenies inferred from the same sequencing data using *mutational signatures*. It was previously shown that distinct mutational processes shape the genomes of the clones comprising a tumor. These processes result in distinct mutational patterns, summarized by a small number of mutational signatures [12, 13]. Current analyses of exposures to mutational signatures do not fully incorporate a tumor's evolutionary context; conversely, current tumor phylogeny estimation methods do not incorporate mutational signature exposures. Here, we introduce the *Tree-constrained Exposure* (TE) problem to infer a small number of exposure shifts along the edges of a given tumor phylogeny. Our algorithm, PhySigs, solves this problem and includes model selection to identify the number of exposure shifts that best explain the data. We validate our approach on simulated data and identify exposure shifts in lung cancer data, including at least one shift with a matching subclonal driver mutation in the mismatch repair pathway. When applying PhySigs to the solution space $\mathcal{T}$ of plausible trees for a single patient, we can then prioritize the trees that have the fewest number of exposure shifts and are therefore more parsimonious with respect to signature exposure. We include an R package along with a tool to allow users to visualize exposure shifts for various trees in a patient solution space.

Chapter 4 likewise introduces a method for prioritizing the solution space for estimated tumor phylogenies for a single patient tumor. Rather than using mutational signatures, here we utilize sequencing data sourced from cohorts of patient tumors. The idea is to leverage the fact that tumors in different patients are the consequence of similar evolutionary processes. We wish to resolve ambiguities in our input data and detect subtypes of evolutionary patterns by simultaneously (i) identifying a single tumor phylogeny among the solution space of trees for each patient, (ii) assigning patients to clusters and (iii) inferring a consensus tree summarizing the identified expanded trees for each cluster of patients. We formalize this as the *Multiple Choice Consensus Tree* (MCCT) problem. We show that this problem is NP-hard via a reduction to 3-SAT and propose a gradient descent heuristic to use in practice. This framework addresses the limitations of previous work in that our problem statement allows for different patient subtypes and our algorithm scales to larger sets of mutations. On simulated data, we show that we are able to better recover the true patient trees and patient clusters relative to existing approaches that do not account for patient subtypes. We then use RECAP to resolve ambiguities in patient trees and find repeated evolutionary trajectories in lung and breast cancer cohorts.

In summary, this dissertation is structured as follows; after introducing key background material in Chapter 2, each chapter introduces a new method for improving phylogeny estimation. Chapter 3 introduces OCTAL, a method for adding missing species into gene trees using a reference tree. Chapter 4 introduces TRACTION, which builds on OCTAL by using a reference tree to also correct low-support branches in gene trees. The next two chapters then shift to improving tumor phylogeny estimation. Chapter 5 presents PhySigs, a method for improving tumor phylogeny estimation with mutational signatures and Chapter 6 presents RECAP, a method for improving tumor phylogeny estimation using cohorts of patient tumors. Chapter 7 concludes with a discussion of future directions.

CHAPTER 2: BACKGROUND

This chapter contains background material referenced throughout this dissertation. Note that key acronyms and terminology are defined once in this section rather than in each subsequent chapter. Sections 2.1–2.2 introduce species phylogenies and tumor phylogenies, respectively. Each section begins with a discussion of the graph-theoretic objects before moving on to current phylogeny estimation models and methods.

2.1 SPECIES PHYLOGENY CONSTRUCTION

Species phylogenies are graphical objects used to represent the evolutionary relationship between species as they have evolved through time. Species phylogenies implicitly assume that the represented species arise from a common ancestor and that genetic material is passed down from parents to children along the branches of the tree. Because this process happened in the past, phylogenies cannot be directly observed, but instead must be estimated. With the advent of high throughput sequencing technology, modern phylogeny estimation methods leverage molecular sequence data sampled across species along with models of evolution to reconstruct this history. In this section, we first introduce species phylogenies as a graphical objects so as to define basic terminology. Next, we describe models of evolution under which phylogenies are estimated and clarify the relationship between two subtypes of species phylogenies: gene trees, estimated over a single locus, and species trees, estimated over the whole genome.[1] Finally, we discuss current methods for species phylogeny estimation along with corresponding limitations.

2.1.1 Species phylogenies as graph theoretic objects

A species phylogeny can be represented as a tree T with leaves labeled by a set S of organisms. If each leaf label is unique, then the species phylogeny is *singly-labeled*. Unless noted otherwise, species phylogenies throughout this dissertation are singly-labeled and unrooted.

Rooted phylogenies. Rooted trees offer an intuitive way to visualize evolution. Indeed, since time is directed, directed edges can be interpreted as capturing the passage of time. However, estimates of species phylogenies are frequently unrooted because roots can be hard

[1] Note that we used the phrase "species phylogeny" to refer to *both* gene and species tree estimation (where the leaves correspond to species) as a way to distinguish these trees from tumor phylogeny estimation.

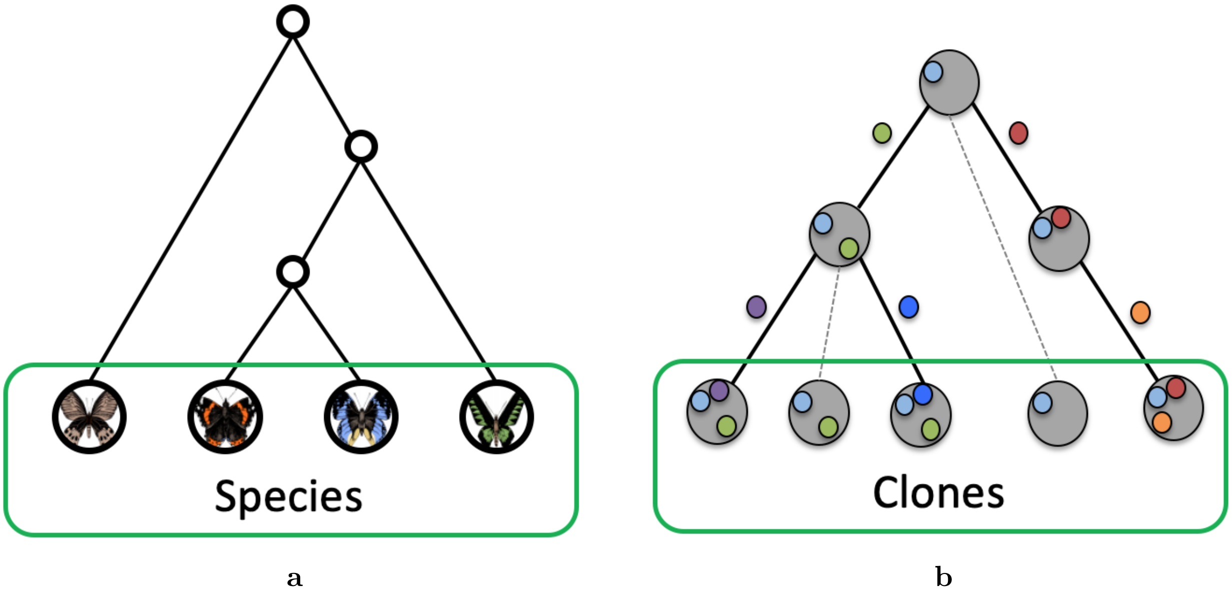


Figure 2.1: **Illustrative examples of the two types of phylogenies in this dissertation.** (a) Example of a rooted species phylogeny where the leaves represent extant species. (b) Example of a tumor phylogeny where the leaves represent extant clones (i.e., cells sharing a common set of mutations). The colored circles denote the introduction of a new mutation, distinguishing a clone from its parent.

to place (see [14] for a discussion). We introduce rooted trees here as a way to motivate the unrooted species phylogenies we work with for the remainder of this dissertation.

A rooted phylogeny can be represented as a rooted tree T with root vertex r. If we direct the edges of this tree away from the root, then each vertex, with the exception of r, will have an indegree of one. When observing a directed edge from u to w between two vertices u, w in this tree, we call u the *parent* of w. Likewise, w is the *child* of u. We denote such an edge with either $u \to w$ or (u, w). More broadly, a vertex u is an *ancestor* of vertex w (and conversely w is a descendant of u) if there exists a directed path from u to w. If a vertex does not have any children, then it is considered to be a *leaf*. Otherwise, the vertex is an *internal node*. In practice, leaves correspond to extant species and internal nodes correspond to ancestral species. Vertices with exactly two children are *binary*, and vertices with more than two children are *polytomies*. A tree where each vertex is binary is called a *binary tree*.

Given a subset of leaves $R \subseteq S$ in tree T, a vertex is a *common ancestor* of R if it is an ancestor of every leaf in R. The common ancestor farthest from the root is known as the *most recent common ancestor* (MRCA) or *lowest common ancestor* (LCA) of R. A *clade* of T is simply a subtree of T; clades correspond to a subset of species defined by a common ancestor and all descendants of that ancestor.

Unrooted phylogenies. A rooted species phylogeny can be converted into an unrooted phylogeny by suppressing the orientation of the edges in the graph and removing any vertex with degree two, connecting its neighbors by a single undirected edge. Each edge e in an unrooted species phylogeny then defines a *bipartition* π_e (also sometimes referred to as a split) on the set of leaf labels induced by the deletion of e from the tree, but not its endpoints. Each bipartition splits the leaf set into two non-empty disjoint parts, A and B, and is denoted by $A|B$. The set of bipartitions of a tree T is given by $C(T) = \{\pi_e : e \in E(T)\}$, where $E(T)$ is the edge set for T. The leaves of an unrooted tree are vertices with degree one, and the remaining vertices are still considered internal nodes. Notice that without directed edges, concepts like parent, child, and clade are no longer well-defined.

Comparing unrooted phylogenies. When working with species phylogenies, it is important to have frameworks for comparing different trees. Here, we describe common relationships between trees as well as define standard distances used to measure topological similarity. These distances may be deployed in the optimization problems used to construct trees or in evaluating the accuracy of estimated trees when ground truth is known.

We start by noting that trees are often restricted to the same set of leaves before making a comparison. More formally, given a species phylogeny T on taxon set S, T *restricted to* $R \subseteq S$ is the minimal subgraph of T connecting elements of R and suppressing nodes of degree two. We denote this as $T|_R$. If T and T' are two trees with R as the intersection of their leaf sets, their *shared edges* are edges whose bipartitions restricted to R are in the set $C(T|_R) \cap C(T'|_R)$. Correspondingly, their *unique edges* are edges whose bipartitions restricted to R are not in the set $C(T|_R) \cap C(T'|_R)$.

We say that a tree T' is a *refinement* of T if T can be obtained from T' by contracting a set of edges in $E(T')$. A tree T is *fully resolved* (i.e., binary) if there is no tree that refines T other than itself. A set Y of bipartitions on some leaf set S is *compatible* if there exists an unrooted tree T leaf-labeled by S such that $Y \subseteq C(T)$. A bipartition π of a set S is said to be compatible with a tree T with leaf set S if and only if there is a tree T' such that $C(T') = C(T) \cup \{\pi\}$ (i.e., T' is a refinement of T that includes the bipartition π). Similarly, two trees on the same leaf set are said to be compatible if they share a common refinement. An important result on compatibility is that pairwise compatibility of a set of bipartitions over a leaf set ensures setwise compatibility [15, 16]; it then follows that two trees are compatible if and only if the union of their sets of bipartitions is compatible.

The *Robinson-Foulds (RF) distance* [11] between two trees T and T' on the same leaf set is defined as the minimum number of edge-contractions and refinements required to transform T into T'. For singly-labeled trees, the RF distance equals the number of bipartitions present in only one tree (i.e., the symmetric difference). We define this formally, as it is frequently used in subsequent chapters.

Definition 2.1. Given singly-labeled trees T and T' both on leaf set S, the *Robinson-Foulds distance* is defined as

$$RF(T,T') := |C(T) \setminus C(T')| + |C(T') \setminus C(T)| \tag{2.1}$$

where $C(T)$ and $C(T')$ are the sets of bipartitions in T and T', respectively.

The normalized RF distance is the RF distance divided by $2n-6$, where n is the number of leaves in each tree; this produces a value between 0 and 1 since the two trees can only disagree with respect to internal edges, and $n-3$ is the maximum number of internal edges in an unrooted tree with n leaves.

The *matching distance* [17] is another distance measure that relaxes the RF distance by giving bipartitions forming a similar partition over the species set partial credit. Formally, given two trees T and T' on the same set of leaves, we define a complete weighted bipartite graph $G = (C(T), C(T'), E)$ such that one set of vertices corresponds to bipartitions in T, and the other set corresponds to bipartitions in T'. The weight of an edge in G between endpoints $\pi \in C(T)$ and $\pi' \in C(T')$ is the minimum hamming distance between any binary vector encoding of each bipartition (i.e., consider an encoding and its complement). The matching distance between T and T' is then defined as the minimum weight perfect matching on G. Note that if the edge weights were instead binary, with 0 for equal endpoints and 1 otherwise, then the minimum weight perfect matching would equal the RF distance.

A common measure that does not directly rely on bipartitions is the *quartet distance* [18]. A *quartet* is defined as the unrooted tree induced when restricting a tree to a subset of four distinct leaves. The quartet distance between two trees T and T' on the same set of leaves is then the number of four taxon subsets for which the quartet differs between trees. The motivation for using quartets is that they represent the smallest set of species for which there is more than one possible unrooted tree topology (in fact, there are three).

2.1.2 Gene and species trees

The precise interpretation of a species phylogeny can vary depending on the biological context. In particular, there are two common subtypes of species phylogenies: gene trees and species trees. Here we introduce these subtypes and discuss the underlying models of evolution that give rise to this distinction.

Gene trees. In the context of this literature, a *gene* is simply defined as a contiguous subsequence of the genome (not necessarily coding for a protein). The evolutionary history of a gene is then represented with a *gene tree* and estimated from genetic sequences descended from some common ancestral gene. This ancestral gene was passed down through many generations, ultimately arriving in the genomes of the individuals labeling the gene tree leaves. For the branching structure to hold, gene trees must be estimated from regions of the genome that are free of *recombination* [10]; in other words, the gene sequence was inherited from one parent in every generation rather than intermixed between different individuals.

As a gene replicates and its copies are passed on, mutations or duplications may occur that generate branching events in the gene tree. When the leaves of a gene tree sample individual representatives of different species, leaves are either related via a speciation event (*orthologs*) or via a duplication event (*paralogs*) (see discussion in [19]). The internal nodes of the gene tree can be interpreted as corresponding to these ancestral events. Note that within a species, individuals may have different versions of a gene, referred to as *alleles*.

Species trees. We begin by acknowledging that there is not a definitive definition of a species, and delineating species is itself an area of research [20]; for the purposes of developing computational methods, we sidestep this issue by assuming that sequencing data from distinct species has already been identified and provided as input.

While a gene tree traces an individual gene lineage as it moves backwards in time, a *species tree* dictates the collection of potential ancestors from which a gene can be inherited. Branches of a species tree can be understood to contain generations of individuals; when a population becomes separated by speciation, the genes within each nascent species likewise become divided into two sets of lineages. In this way, gene trees are constrained by branches of the species tree [10]. Internal nodes of a species tree therefore correspond to *speciation* events, lineage-splitting that creates two or more separate species. We typically expect for an ancestral species to diverge into two descendant species, creating a binary species tree. However, there are some examples of ancient *rapid radiation events*; these are short periods of intense diversification that are especially hard to resolve, creating polytomies in the species tree [21]. Unlike gene trees estimated from a single locus, species trees incorporate sequencing data from across the genome.

Gene and species tree discord. Somewhat surprisingly, certain biological processes can cause the topologies of gene trees to differ from each other as well as from the species tree (see Fig. 2.2). Three major causes of this discord were famously identified in [10], and we will describe each of them here. First, *incomplete lineage sorting* (ILS) refers to a population-level phenomenon whereby gene lineages coalesce in a branch of the species tree deeper than their MRCA. When this happens, the ordering of lineage merging may be such that it differs from the species tree. ILS is thought to be widespread and found across a variety of species (e.g., [22, 23, 24]). *Gene duplication and loss* (GDL) refers to the process whereby genes duplicate to create additional copies or are deleted during replication [25]. Complex patterns of duplication and loss can likewise lead to a gene tree that differs from the species tree. GDL has been observed in nature, including between the human and great ape [26]; in fact, gene duplication is thought to be a major driver of evolutionary change as the duplicated gene takes on new functions (the orthologs conjecture). Finally, *reticulate evolution* describes the merging of ancestral lineages into a descendent lineage. Examples of reticulate evolution include hybridization [27] along with horizontal gene transfer (HGT) [28], where an organism incorporates genetic information from a contemporary organism other than its parent. Reticulate evolution is well documented in bacteria and single cell eukaryotes (e.g., [29, 30, 31, 32]), but has also been observed in larger organisms such as plants (e.g., [33, 34, 35]).

There is a major conceptual difference between reticulate evolution as compared to ILS and HGT. The merging of branches in reticulate evolution results in a non-treelike process and requires a more general graphical model, called a *phylogenetic network*, for proper analysis [36, 37, 38, 39]. However, GDL and ILS produce heterogeneity across the genome that can still be properly modeled by a single species tree [10, 40]. Species phylogeny estimation methods should account for and be robust to this heterogeneity.

Models relating gene and species trees. Models of evolution allow us to establish statistical relationships between gene and species trees; as motivated above, they are not wholly independent. Indeed, a gene tree can be thought of as randomly drawn from the gene tree distribution defined by a species tree. Conversely, a species tree can be inferred using probabilistic models accounting for the distribution of gene trees induced by species trees.

Much of the focus in the statistical phylogenetics literature has been on developing methods for species tree estimation in the presence of ILS, which is modeled by the *multi-species coalescent* (MSC) [41, 42, 43, 44]. Under the MSC model, each branch of the species tree is parameterized by the number of generations spanned by the branch and the size of the population contained in the branch (or twice that if species are diploid). These parameters are the branch length and width, respectively. Since the probability that two lineages coalesce on a branch is simply a function of length and width, the parameterized species tree defines a probability distribution over possible gene trees contained within its branches. Species trees with very short branches or very large populations are more likely to produce conflict-

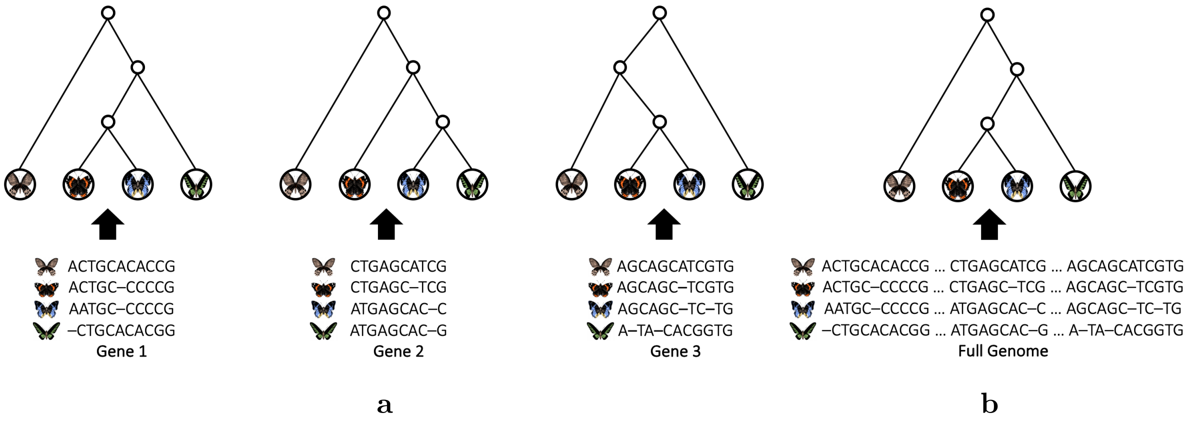

ACTGCACACCG
ACTGC–CCCCG
AATGC–CCCCG
–CTGCACACGG
Gene 1
CTGAGCATCG
CTGAGC–TCG
ATGAGCAC–C
ATGAGCAC–G
Gene 2
AGCAGCATCGTG
AGCAGC–TCGTG
AGCAGC–TC–TG
A–TA–CACGGTG
Gene 3
ACTGCACACCG ... CTGAGCATCG ... AGCAGCATCGTG
ACTGC–CCCCG ... CTGAGC–TCG ... AGCAGC–TCGTG
AATGC–CCCCG ... ATGAGCAC–C ... AGCAGC–TC–TG
–CTGCACACGG ... ATGAGCAC–G ... A–TA–CACGGTG
Full Genome
a
b

Figure 2.2: **Heterogeneity in modeling the evolution of single gene versus entire genome.** (a) Examples of rooted gene trees, which depict the evolutionary history of a single gene for a collection of butterfly species. Note that tree topologies may differ across genes due to certain known biological phenomena, such as GDL, ILS, and HGT. (b) Example of a species tree which depicts the evolutionary history of the entire genome for a collection of butterfly species. Note that the species and gene tree topologies again may differ. In some cases, the species tree may not have the same topology as any gene tree.

ing gene trees. Under the MSC, species trees are provably identifiable from the probability distribution of gene tree topologies they define for certain models of sequence evolution [45]. There has been recent interest in expanding beyond MSC to model types of discord other than ILS. Notable examples of this include a probabilistic model accounting for just GDL [46] followed by a unified model that accounts for both ILS and GDL [47].

Models of evolution are important to keep in mind when thinking about method assumptions and the data on which methods are tested. Simplistic models of evolution can be more computationally tractable but may, for example, suffer from being underspecified, resulting in a large solution space. On the other hand, highly parameterized methods may perform well on data consistent with the model assumptions, but may not be robust or transferable to a wide range of datasets.

2.1.3 Estimation methods

We now discuss existing approaches for both gene and species tree estimation. While the subsequent chapters of this dissertation primarily focus on improving gene trees, we introduce here how gene trees may be used as an input to estimate species trees. In this way, improving gene tree estimation may improve species tree estimation, albeit indirectly.

Evaluating methods. Computational methods must first and foremost perform well empirically to be useful to biologists. Since we do not typically know the ground truth on real biological datasets, simulated datasets are constructed under different models of evolution to benchmark method accuracy. Researchers test methods under a range of conditions, evaluating robustness to error, model misspecification, and challenging edge cases.

It is likewise important to establish the theoretical proprieties of each method. For instance, there is a lot of interest in showing methods are *statistically consistent*; that is, error in the estimated model parameters provably goes to zero as the amount of input data generated under the model approaches infinity. Otherwise, the method is said to be *statistically inconsistent*. While statistical consistency is important, it is worth reiterating that it only describes performance at the limit. In practice, methods are run on finite data, which statistical consistency does not address. Another important theoretical consideration is time complexity, which indicates the scalability of a method to large datasets with many species. Algorithms will need to accommodate large-scale sequencing projects underway such as the 5000 Insect Genomes Project [1, 2], the 10000 Plant Genomes Project [3], and the Earth BioGenome Project, whose goal is to sequence 1.5 million eukaryotic species [4].

Gene tree estimation. The input to gene tree estimation methods is typically a *multiple sequence alignment* (MSA) for the gene of interest across a set of species. An alignment can be represented as a matrix where each row contains the genomic sequence for a species interspliced with *gap* character states; gaps are introduced so as to ensure that characters in the same column are *homologous*, meaning they evolved from the same nucleotide in a common ancestor. MSAs are an active area of research and beyond the scope of this dissertation; we refer the interested reader to the chapter on MSA in [48].

Given an MSA, there are broadly two types of approaches for estimating a gene tree. Non-statistical approaches include constructing trees from qualitative characters (e.g., maximum parsimony and maximum compatibility) and certain types of distance methods (e.g., neighbor joining). Statistical approaches, on the other hand, explicitly account for models of evolution. Examples of these approaches include maximum likelihood and Bayesian methods. Across methods, there are different trade-offs: some are heuristics for NP-hard problems, some are computationally intensive, some lack certain theoretical properties such as statistical consistency. However, one limitation that all these methods have in common is that gene tree estimation based on a single locus is more susceptible to (1) missing sequencing data for that gene for certain species and (2) not containing enough signal to determine the correct gene tree topology with high confidence (see Chapter 1 and discussion in [49, 50]).

One approach from the GDL literature to address limited signal is to modify estimated gene trees with respect to a reference tree, which may either be an established tree from prior studies or an estimated species tree (e.g., based on an assembled multi-gene dataset). Such methods are typically based on parametric models of gene evolution and broadly fall into two categories. *Integrative methods* use available sequence data in addition to the estimated gene tree and reference tree; examples include ProfileNJ [51], TreeFix [52], and TreeFix-DTL [53]. On the other hand, *gene tree correction methods*, only use the gene tree and species tree topologies; Notung [54, 55] and ecceTERA [56] are two well-known methods of this type. Integrative methods are generally expected to be more accurate than gene tree correction methods in the presence of GDL, but are also more computationally intensive as a result of performing likelihood calculations with the sequencing data. See [57, 58, 59, 60, 61, 62] for additional literature on this subject. We build on this line of work in Chapters 3-4 by introducing non-parametric approaches for gene tree correction that can be used in contexts outside of GDL, such as ILS and HGT.

Species tree estimation. A straightforward approach for estimating species trees is simply to concatenate together a multi-locus dataset and then run a method described above on the resulting alignment. Such an approach is often referred to as *concatenation* and is indeed quite popular [63, 64]. Despite competitive empirical performance on benchmark experiments [65], these methods tend to suffer from a number of theoretical limitations. For example, one of the most common versions of concatenation uses maximum likelihood (CA-ML). Under the simplest unpartitioned version of CA-ML, all loci are assumed to evolve down a single model tree, an assumption that is violated when recombination occurs between loci. Furthermore, it was shown that unpartitioned CA-ML is statistically inconsistent and sometimes positively misleading under the MSC model [44].

Another intuitive approach for estimating species trees from multi-locus data is to first estimate gene trees and then combine them into a species tree, leveraging the fact that the two types of trees are related. The simplest way to combine them is to use the most frequent gene tree topology as the estimate of the species tree. This approach has been proven statistically consistent under the MSC for rooted species trees with three leaves or unrooted trees with four leaves, but not when more leaves are present [66, 67, 68, 69]. However, there are related approaches that combine gene trees into a larger species tree in a way that is statistically consistent under the MSC model. Some of these *summary methods*, such as ASTRAL-II [70] and ASTRID [71], have been shown to scale well to datasets with many taxa (i.e., >1000 species) and provide accurate species tree estimates. Note that summary methods have many features in common with *supertree methods* that combine source trees on overlapping leaf sets (see discussion in [72]), but are based on mathematical properties of the MSC model and so can be proven statistically consistent under this model; supertree methods, by contrast, assume conflict between source trees is due to estimation error rather than ILS, and so are generally not statistically consistent under the MSC model.

Finally, there also exist *co-estimation methods* that indirectly rely on gene trees. These methods work by taking as input an MSA and then inferring gene and species trees together [73]. While these methods are often computationally expensive, the approach supports the idea that species and gene trees may be able to improve one another iteratively.

2.2 TUMOR PHYLOGENY CONSTRUCTION

We now switch our attention from species phylogenies to tumor phylogenies. Tumor phylogeny estimation is a related but distinct conceptual challenge from species phylogeny estimation. Many of these differences stem from the fact that evolution is happening at a different scale, where the leaves of the phylogeny are not known a priori.

Tumors develop via an evolutionary process [5]; tumor cells rapidly grow and divide, acquiring new mutations with each subsequent generation. Mutations that accumulate during the lifetime of an individual are referred to as *somatic mutations* as opposed to *germline mutations* that are inherited. Under the *infinite sites assumption* (ISA) each mutation is gained exactly once and never lost, giving rise to a two-state *perfect phylogeny.* The ISA underlies the majority of current methods for tumor phylogeny inference from both bulk and single-cell DNA sequencing data (reviewed in [74]). In line with this work, we likewise adhere to the ISA unless it is noted otherwise; we acknowledge that this is a simplifying assumption that should be relaxed in future work.

2.2.1 Tumor phylogenies as graph theoretic objects

We begin by reviewing the basic graphical objects underlying tumor phylogenies. Note that we hold off on diving into tumor biology until the next section; as a consequence, certain terms like mutation are left intentionally vague.

Rooted phylogenies. The evolutionary history of n mutations of a tumor is represented by a particular type of rooted tree T whose root vertex is denoted by $r(T)$, vertex set by $V(T)$, and directed edge set by $E(T)$. Similar to the rooted species phylogenies introduced above, the edges of a tumor phylogeny are directed away from the root. The terms *parent*, *child*, *ancestor*, and *descendant* are used in an analogous fashion when describing the relationship

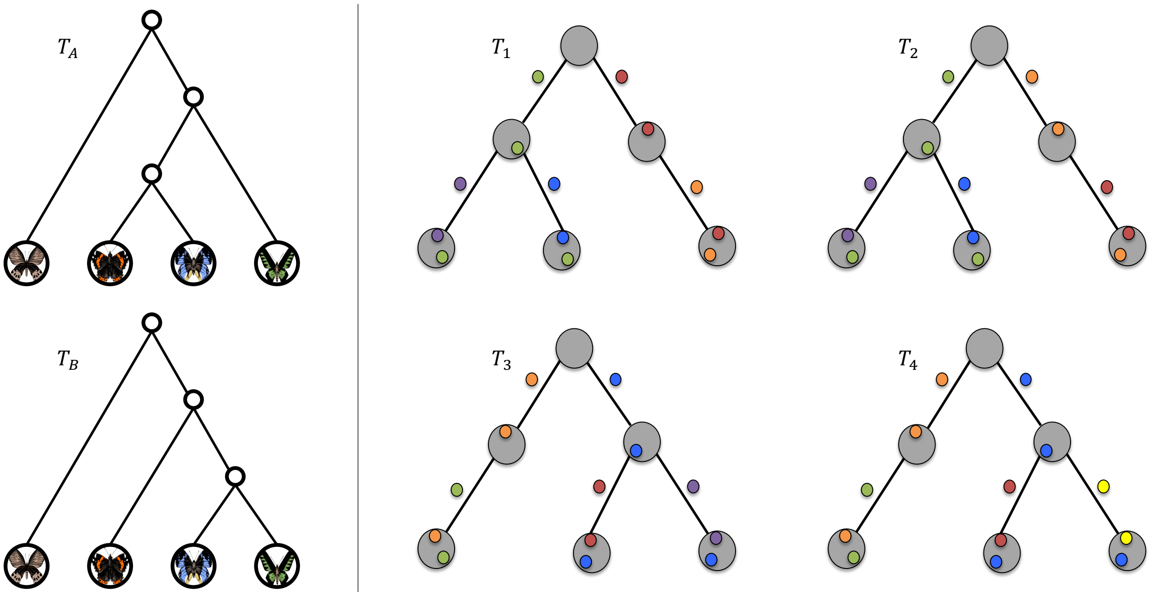


Figure 2.3: **Unlike a species phylogeny, leaves of a tumor phylogeny are not known a priori.** (Left) Two different species phylogenies T_A, T_B with the same set of species labeling the leaves. (Right) Four different tumor phylogenies that differ in distinct ways. Trees T_1 and T_2 have the same set of clones labeling the leaves, but a different ancestral relationship between the mutations on the right branch. Trees T_1 and T_3 have the same set of mutations but different clones. Trees T_1 and T_4 have different mutations and different clones.

between two vertices. Other general tree terminology also carries over directly, including *binary*, *polytomy*, *common ancestor*, and *most recent common ancestor*.

The vertices of T now represent groups of cells, or *clones*, that have nearly identical mutational profiles. The leaves of T correspond to the extant clones present at the time of sampling, and the internal nodes correspond to ancestral clones in the tumor. Under the ISA, mutations are gained once and never lost. Thus, each mutation is present on exactly one edge of T, and we may label each non-root vertex $v \neq r(T)$ by the mutations $\mu(v) = \mu(u, v)$ introduced on its unique incoming edge (u, v). The root vertex $r(T)$ typically corresponds to the normal, or non-mutated clone, and is represented by the empty set $\mu(r(T)) = \emptyset$.

If more than one mutation is gained on an edge, then we call the child vertex v a *mutation cluster* (i.e., $|\mu(v)| > 1$). Such mutation clusters represent a type of ambiguity where the linear ordering of mutation gain is unknown. We say that a tree T' is an *expansion* of a tree T if all mutation clusters of T have been expanded into ordered paths such that only one mutation is gained on each edge. In particular, for a mutation cluster with n mutations, there are $n!$ possible expanded paths. We also define the *ancestral set* $A(v)$ of vertex v as the set of mutations that label the path from $r(T)$ to v in T. $A(v)$ defines the full set of mutations

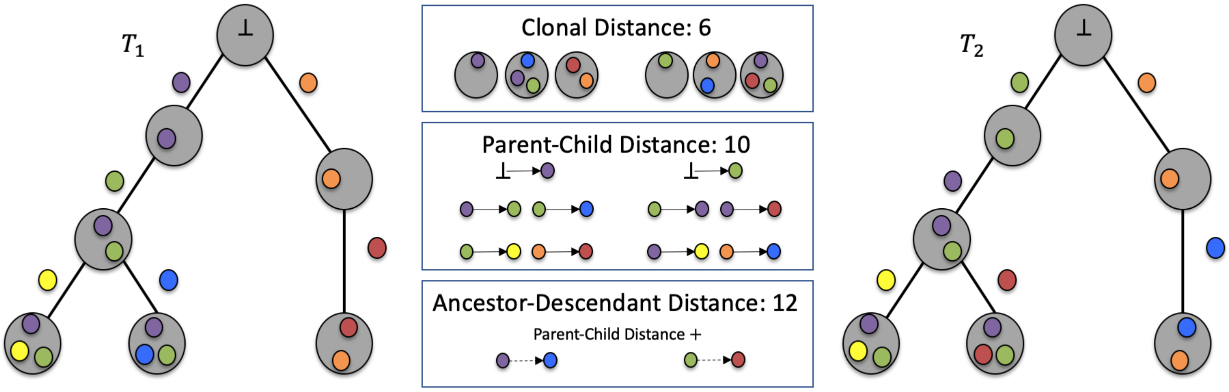


Figure 2.4: **Three different distance measures for comparing tumor phylogenies.** Each measure takes the symmetric difference between sets from each tree. Clonal distance uses the set of clones, parent-child distance uses the set of directed edges, and ancestor-descendant distance uses the set of all ancestor-descendant mutation pairs. Here, we compute each distance between trees T_1 and T_2. We show the elements contributing to the symmetric difference in each case. Items in T_1 but not T_2 are shown in the column on the left. Conversely, items in T_2 but not T_1 are shown in the right column.

present in the clone represented by v. We say that two edges $(u, v) \in E(T)$, $(u', v') \in E(T')$ are equal if they have the same set of mutations present at their corresponding endpoints (i.e. $\mu(u) = \mu(u'), \mu(v) = \mu(v')$).

Comparing rooted phylogenies. There are several reasons why the methods for comparing rooted species phylogenies do not immediately translate to tumor phylogenies. The first reason is that the leaves of a tumor tree are not known a priori; we typically observe the set of mutations for a patient, but the clustering of these mutations into clones is not directly observed. This creates scenarios where, even for a fixed set of mutations, we may want to compare alternative tumor phylogenies with different clones. A second reason is that there is only one tree of life, but there are many separate instances of tumor evolution occurring in different patient tumors. Here, we may want to compare trees not only with different clones, but also with different sets of mutations. In other words, tumor comparisons must operate at the level of the mutation sets labeling each vertex rather than collapsing these sets into some higher order notion of species (see Fig. 2.4).

Recall that above we introduced the notion of RF distance between unrooted species phylogenies. In the rooted setting, RF distance can be modified slightly to equal the symmetric difference of the sets of clades rather than bipartitions. In an attempt to adapt this to tumor phylogenies, we could think of a clade as being represented by the set of mutations gained within that clade. However, there is something flawed with this approach; working with clades emphasizes branched clones rather than clones with an ancestral relationship. We are primarily interested in ancestral relationships since ancestral mutations influence the occurrence of future mutations. We can instead look at something called *clonal distance* [75], which is the symmetric difference of the clones (i.e., ancestral sets) between two tumor phylogenies. In other words, rather that looking at the set of mutations gained below a vertex, we look above at the set of mutations gained on the path from vertex to root. A challenge with this method is that two clones that are nearly identical are not distinguished from clones that have no mutations in common. In practice, trees often have no clones in common, a case where this measure has little utility.

One of the most common distances used in the literature is the *parent-child* (PC) distance, which is equal to the symmetric difference in the edge sets between two trees [75]. We define it formally here as it is referenced directly later. This distance only captures the dependency of a mutation on the preceding mutation, which has been criticised as a limitation for a similar reason as above; all inherited mutations on the path to the root may play a role in the selection of the next mutation, not just the most recent one. It is also a concern that, in practice, not all mutations are sampled from every tumor, and so the parent-child distance may be too sensitive of a measure.

Definition 2.2. The *parent-child distance* $d(T, T')$ of two trees T and T' is the size of the symmetric difference between the two edge sets $E(T)$ and $E(T')$, i.e.

$$d(T, T') = |E(T) \bigtriangleup E(T')|. \tag{2.2}$$

The *ancestor-descendent* (AD) distance is a slightly more generalized measure that incorporates long-range relationships. Given two trees T and T', this distance is computed by completing each tree with all edges between ancestor-descendent pairs. The distance is then the symmetric difference between the completed edge sets (i.e., the number of ancestor-descendant pairs in one tree but not the other) [75]. This approach corrects for some of the shortcomings of parent-child distance, but can introduce other unintended artifacts. For instance, two trees with the same number of mutations can now have very different numbers of ancestor-descendent pairs because of differences in tree depth. Certain applications that rely on this measure might inadvertently weight trees with larger depth more heavily.

Many other measures have been proposed for comparing tumor phylogenies under the ISA [76, 77, 78], and this continues to be an area of ongoing research. It is likely that no one measure will be appropriate across contexts, and it is important to understand the trade-offs when incorporating these measures into optimization functions or evaluating results.

2.2.2 Clonal theory of evolution

The landmark paper by Nowell [5] posits that cancer results from an evolutionary process whereby cells within a tumor descend from a single *founder cell*. As cells grow and divide, new mutations are introduced leading to genetically-distinct subpopulations of cells known as clones. Clones are subject to selective pressure exerted by surrounding tissues, the immune system, and treatments. Particularly advantageous combinations of mutations are selected for and create *clonal expansions*, where many copies of a particular clone arise in a tumor.

Types of somatic mutations. Genetic variation observed in cancer can be classified into different types depending on the scale in which it operates. Small-scale variations ($<$ 1 kilobase in length) include insertions and deletions as well as point mutations. We note that when germline point mutations are common to a population, they are referred to as *single nucleotide polymorphisms* (SNPs). This is not to be confused with somatic point mutations arising in tumor cells, which are referred to as *single nucleotide variants* (SNVs).

On the other hand, large-scale variations ($>$ 1 kilobase in length) include chromosomal rearrangements (e.g., translocation, transversion, segmental) and copy number aberrations (CNAs), such as gain or loss. At the largest scale, there can be variation in the number of whole chromosomes or genomes (e.g., polyploidy, aneuploidy).

Models of tumor evolution. Much of the current work in tumor phylogenetics focuses on just SNVs and relies on the ISA. Recall, under the ISA each mutation can only be gained once and never lost. In the two-state case, this corresponds to a tree where the root vertex starts with state 0 (non-mutated) for each character. Each character can change state from 0 to 1 only once in the tree, but never revert back. This model is known as the *two-state perfect phylogeny* model. A more general version has also been proposed, called the *multi-state perfect phylogeny* model, which likewise prohibits homoplasy. A character in this model can change states more than once, but never change back to a previous state.

SNVs and CNAs can interact in such a way that reliably violates the standard ISA assumption, making the two-state perfect phylogeny model unrealistic. For instance, a copy number deletion that overlaps with an SNV causes a loss of that SNV in descendent cells. One straightforward solution that is often deployed is to exclude all SNVs that occur in regions where a CNA has been detected. However, aneuploidy drives extensive CNAs in approximately 90% of tumors [79, 80]; removing these regions results in a major loss of data. More general models of evolution have been proposed in response to these concerns. While the methods in this dissertation are primarily developed under ISA, we include these models because they are an important direction of future research.

Under the *Dollo parsimony model* [81], a mutation can be gained only once, but it can be lost multiple times. A variation on this model is the *k-Dollo parsimony model*, which restricts each mutation to being lost at most *k* times [82]. Because the loss of SNVs due to CNAs is much more prevalent than gaining the same mutation twice via parallel evolution, this relaxation captures dominant factors underlying the evolution of SNVs in cancer.

The *finite sites model* is even more general, allowing for parallel evolution along with mutation loss. This is a parameterized model that uses a continuous-time Markov chain to assign probabilities along branches of the tumor phylogeny for each possible transition between a homozygous reference or a heterozygous or homozygous non-reference genotype. The model assumes that each site in the genome evolves identically and independently through time. Thus, a major conceptual limitation of both this and the Dollo model is that although they allow for loss, they do not consider CNAs that produce loss dependencies across sites. Recent work [83] has tried to address this concern by using a loss-supported phylogeny model, which constrains losses to regions with evidence of copy number decrease.

Hallmarks of cancer. While each tumor results from a different instantiation of this evolutionary process, it is postulated that the complexity of all cancers can be reduced to a small number of principles, so called *hallmarks of cancer* [84, 85]. The theory states that normal cells must obtain a certain set of traits, such as "evading growth suppressors," "resisting cell death," and "enabling replicative immortality," to eventually turn malignant. The increasing availability of tumor sequencing data has led to the use of phylogenies to identify *driver mutations* interfering with genes or pathways potentially linked to these common traits driving cancer progression [86, 87].

Driver mutations, in turn, may then be used to identify repeated *evolutionary trajectories* in tumorigenesis and metastasis [88, 89, 90, 91]. Crucially, it has been suggested that the *co-occurrence* and *ordering*, rather than just mutation presence, have significant prognostic consequences, as was shown, for example, in a study of myeloproliferative neoplasms with the ordering of JAK2 and TET2 mutations [92]. Early methods for revealing evolutionary trajectories from cohorts of patients find evidence of reoccuring patterns, but suffer from scalability issues and make simplifying assumptions about patient subtypes [90, 91]. It is the hope that identifying patient subtypes with shared evolutionary patterns, called *evolutionary subtypes*, will eventually lead to clinically-relevant subtypes enabling precision medicine. Nevertheless, there is an exponential number of orderings in which these mutations can be acquired and identifying these hallmarks remains a great challenge. We address this issue further in the final chapter of this dissertation.

2.2.3 Estimation methods

Tumor phylogeny estimation methods have been developed for a range of sequencing technologies. However, most current methods only reconstruct the evolutionary history of SNVs, with some methods also accounting for changes in copy number [82, 93, 94]. We focus on such methods here unless otherwise noted. Future methods will need to account for the other types of structural variation that can occur in the genome.

Bulk sequencing. The majority of patient tumor data is currently obtained via bulk DNA sequencing. For instance, almost all datasets from The Cancer Genome Atlas (TCGA) and International Cancer Genome Consortium (ICGC) consist of single bulk tumor samples. The input to bulk sequencing is a mixed sample of short reads ($<$ 300 bp) taken from potentially millions of cells with varying genomes [95, 96]. In practice, samples will not just contain tumor cells, but also normal cells. The fraction of tumor cells in a sample is known as the *purity*. Reads are then aligned to a reference genome. For each SNV present in a bulk sample, we directly observe the fraction of DNA sequencing reads aligned to that location in the genome containing a variant allele. This fraction is the *variant allele frequency* (VAF), and the denominator (i.e., the number of aligned reads) is the *read depth*. The VAF estimates the fraction of tumor chromosomes containing an SNV with some error introduced by the stochasticity of sequencing. Note that this is not the same as the fraction of cells containing an SNV, known as the *cancer cell fraction* (CCF), as purity and copy number obscure the relationship. However, VAF can be assumed as proportionate to CCF under the ISA if CNAs are ignored since this implies all mutated cells are heterozygous diploid. Alternatively, several methods have been developed to use copy number to more carefully convert VAF into CCF [97]. Tumor phylogeny estimations methods then use VAF or CCF to simultaneously identify the clones and tree topology over these clones (see [74] for a survey).

Methods here are inherently trying to solve a mixture problem; intuitively, we have as input a matrix F, where the rows correspond to tumor samples and columns correspond to mutations. The entries of this matrix are the proportion of cells containing a mutation in a sample. Informally, the objective is to factorize F into (1) a matrix B specifying the clustering of mutations into clones, and (2) a mixture matrix U describing the proportion of clones in each sample. When additional constraints are placed on B so that the clones are consistent with an evolutionary model, we call it a phylogeny mixture problem. Much work has been done on studying the *Perfect Phylogeny Mixture* (PPM) problem [98], where the evolutionary model placed on B is the two-state perfect phylogeny (i.e., B must correspond to a perfect phylogeny). Deciding if such a factorization of F into UB exists in this setting has been shown to be NP-complete [98]. Minimizing the factorization error or sampling from the solution space is therefore NP-hard; note that the solution is not necessarily unique [99].

Several heuristic methods have been developed that try to solve variations of this mixture problem in practice. Methods that specialize in estimating tumor phylogenies from just one bulk sample typically need to make additional strong assumptions, such as strong parsimony or sparsity [100], because one sample does not provide sufficient constraint. Fortunately, as sequencing has become cheaper and more pervasive, there has been a rise in bulk sequencing studies that collect multiple samples from the same tumor, either spatially (e.g., [87, 101, 102, 103]) or temporally (e.g., [89, 104]). This has led to a boon in multi-sample bulk sequencing methods, some of which just model SNVs and others that also account for CNAs [98, 105, 106, 107, 108]. A method has also been proposed that moves beyond two-state and assumes the multi-state perfect phylogeny model [109]. A current limitation shared by all of these methods is that they do not infer a single tree per patient but rather a large solution space of plausible trees for each individual patient [95, 96]. Identifying the true tree is important to effective downstream analysis; either additional data must be collected or additional constraints should be explored in order to select one tree for each patient.

Single cell sequencing. Single-cell sequencing allows us to directly observe specific cells present within a tumor without deconvolution. While promising, single-cell sequencing is currently expensive and not performed at scale. Moreover, the data suffers from a number of technical challenges that must be adequately addressed. These sequences are known to have elevated rates of false positives and negatives from genome amplification, as well as significant amounts of missing data since only a small fraction of tumor cells are sequenced [110, 111]. This is especially problematic in the context of cancer, where it is hard to determine if a variant is a subclonal mutation or an error. While early single cell methods directly applied classic phylogeny estimation techniques like neighbor joining [112], more recent methods have tried to explicitly model the single cell sequencing errors for better results [77, 82, 93, 113, 114]. Still, single cell methods have some limitations, especially in establishing linear mutation orderings, due to high noise levels and missing ancestral clones [115].

Some methods look to combine single cell and bulk sequencing data in a complementary way to overcome the limitations of each technology [115]. For example, bulk sequencing is an inexpensive way to generate a set of plausible tumor phylogenies for a patient. Targeted single cell sequencing can then be used to rule out phylogenies and hopefully select the true tree [116]. While this is again another promising direction, such combined datasets are rare and require the ability to do additional sequencing. Later, we look to see what signal we can extract from existing datasets when additional sequencing is not an option.

CHAPTER 3: CORRECTING GENE TREES TO INCLUDE MISSING SPECIES USING A REFERENCE TREE WITH OCTAL

We begin the main body of this dissertation with a method for adding missing species into gene trees using a reference tree. This reference tree is typically estimated from a multi-locus dataset and used as a surrogate in the case where sequencing information is missing for the gene of interest. Figures and tables appear at the end of this chapter in Section 3.6.

3.1 INTRODUCTION

A common challenge for phylogeny estimation methods is that sequence data may not be available for all genes and species of interest, creating conditions with missing data [49, 50, 119]. Gene trees can be missing species simply because some species do not contain a particular gene, and in some cases, no common gene will be shared by every species in the set of taxa [120]. Additionally, not all genomes may be fully sequenced and assembled, as this can be operationally difficult and expensive [50, 121].

Missing gene data has both practical and theoretical implications, the impact of which may propagate into downstream analysis, including species tree estimation. Although species tree summary methods are statistically consistent under the MSC model [122], the proofs of statistical consistency assume that all gene trees are complete, and so may not apply when the gene trees are missing taxa. Recent extensions to this theory have postulated that some species tree estimation methods are statistically consistent under some models of missing data (e.g., when every species is missing from each gene with uniform probability) [123, 124, 125]. However, missing data in biological datasets often violates such models [119]; for example, missing data may be biased towards genes with faster rates of evolution [126]. Furthermore, sparse multi-locus datasets can be *phylogenetically indecisive*, meaning more than one tree topology can be optimal, making it impossible to distinguish between multiple alternative trees [127]. Because of concerns that missing data may reduce the accuracy of multi-locus species tree estimation methods, many phylogenomic studies have restricted their analyses to only include genes with most of the species (see discussion in [49, 50, 128]).

We address this challenge by adding missing species back into gene trees using auxiliary

This chapter contains material previously presented at the 2017 *Workshop on Algorithms in Bioinformatics* [117] and later published in *Algorithms for Molecular Biology* under the title "OCTAL: Optimal Completion of Gene Trees in Polynomial Time" [118]. This work was done in conjunction with E.K. Molloy, P. Vachaspati, and T. Warnow; TW conceived the project; PV, EKM designed and implemented the algorithm; SC, EKM, TW established theory; SC, TW wrote proofs; SC, PV, EKM performed experiments; SC, EKM produced the figures; SC, EKM, TW contributed to the writing of this chapter.

information from other regions of the genome. In doing so, we formulate the *Optimal Tree Completion* (OTC) problem, which seeks to add missing species to a gene tree so as to minimize distance to another tree, called a *reference tree*. In this chapter, we specifically address the Robinson-Foulds (RF) Optimal Completion problem, which seeks a completion that minimizes the RF distance between the two trees. We then present the Optimal Completion of Incomplete gene Tree Algorithm (OCTAL), a greedy polynomial time algorithm that we prove solves the RF Optimal Completion problem exactly. We also present results from an experimental study on simulated datasets comparing OCTAL to a heuristic for gene tree completion within ASTRAL-II.

3.2 PROBLEM STATEMENT

The problem we address in this paper seeks to add leaves into an incomplete tree in such a way as to minimize distance to an existing tree, where the distance between trees is defined by the RF distance (see Def. 2.1). A formal statement of the problem is as follows:

Problem 3.1 (RF Optimal Tree Completion (RF-OTC))**.** Given an unrooted binary tree T on the full taxon set S and an unrooted binary tree t on a subset of taxa $R \subseteq S$ output an unrooted binary tree T' on the full taxon set S with two key properties:

1. T' is a *S-completion* of t (i.e., T' contains all the leaves of S and $T'|_R = t$) and
2. T' minimizes the RF distance to T among all *S-completions* of t

Note that t and $T|_R$ are both on taxon set R, but need not be identical. In fact, the RF distance between these two trees is a lower bound on the RF distance between T and T'.

3.3 METHODS

In this section, we present OCTAL, an algorithm for solving the RF-OTC problem exactly in polynomial time. The algorithm begins with input tree t and adds leaves one at a time from the set $S \setminus R$ until a tree on the full set of taxa S is obtained. To add the first leaf, we choose an arbitrary taxon x to add from the set $S \setminus R$. We root the tree $T|_{R\cup\{x\}}$ (i.e., T restricted to the leaf set of t plus the new leaf being added) at x, and then remove x and the incident edge; this produces a rooted binary tree we will refer to as $T^{(x)}$ that has leaf set R.

We perform a depth-first traversal down $T^{(x)}$ until a shared edge e (i.e., an edge where the clade below it appears in tree t) is found. Since every edge incident with a leaf in $T^{(x)}$

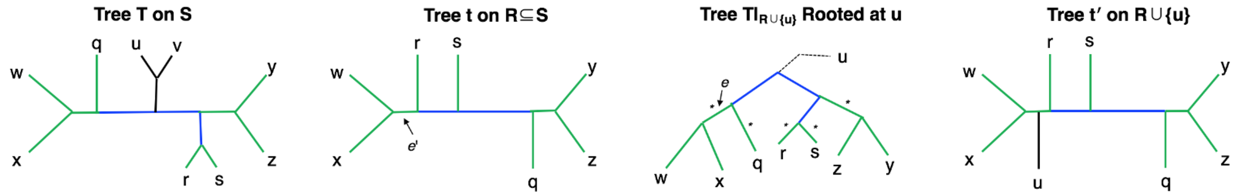


Figure 3.1: **One iteration of the OCTAL algorithm.** Trees T and t with edges in the backbone (defined to be the edges on paths between nodes in the common leaf set) colored green for shared, and blue for unique; all other edges are colored black. After rooting $T|_R$ with respect to u, the edges in $T|_R$ that could be identified by the algorithm for "placement" are indicated with an asterisk (*). Note that any path in $T|_R$ from the root to a leaf will encounter a shared edge, since the edges incident with leaves are always shared. In this scenario, the edge e above the least common ancestor of leaves w and x is selected; this edge defines the same bipartition as edge e' in t. Hence, AddLeaf will insert leaf u into t by subdividing edge e', and making u adjacent to the newly added node.

is a shared edge, every path from the root of $T^{(x)}$ to a leaf has a distinct first edge e that is a shared edge. Hence, the other edges on the path from the root to e are unique edges.

After we identify the shared edge e in $T^{(x)}$, we identify the edge e' in t defining the same bipartition, and we add a new node $v(e')$ into t so that we subdivide e'. We then make x adjacent to $v(e')$. Note that since t is binary, the modification t' of t that is produced by adding x is also binary and that $t'|_R = t$. These steps are then repeated until all leaves from $S \setminus R$ are added to t. This process is shown in Figure 3.1 and given in Algorithm 3.1 below.

3.3.1 Proof of Correctness

Let T be an arbitrary binary tree on taxon set S and t be an arbitrary binary tree on taxon set $R \subseteq S$. Let T' denote the tree returned by OCTAL given T and t. We set $r = RF(T|_R, t)$. As we have noted, OCTAL returns a binary tree T' that is an S-completion of t. Hence, to prove that OCTAL solves the RF Optimal Tree Completion problem exactly, we only need to establish that $RF(T, T')$ is the smallest possible of all binary trees on leaf set S that are S-completions of t. While the algorithm works by adding a single leaf at a time, we use two types of subtrees, denoted as *superleaves*, to aid in the proof of correctness.

Definition 3.1. The *backbone* of T with respect to t is the set of edges in T that are on a path between two leaves in R.

Definition 3.2. A *superleaf* of T with respect t is a rooted group of leaves from $S \setminus R$ that is attached to an edge in the backbone of T. In particular, each superleaf is rooted at the

Algorithm 3.1: RF Optimal Tree Completion Algorithm (OCTAL).

```
Input: Binary tree $t$ on taxon set $R \subseteq S$, binary tree $T$ on taxon set $S$
Output: Binary tree $t'$ on taxon set $S$ solving the RF-OTC problem

Function AddLeaf:
    Root $T_2$ at $v$, the neighbor of $x$, and delete $x$ to get a rooted version of $T_2|_K$
    Pick arbitrary leaf $y$ in $T_2|_K$ and find first edge $e \in E$ on path from $v$ to $y$
    Find $e'$ in $T_1$ defining the same bipartition as $e$
    Attach $x$ to $e'$ in $T_1$ by subdividing $e'$ and making $x$ adjacent to the newly
      created node; call the resulting tree $T_1'$
    return $T_1'$

Function Main:
    if $R$=$S$ then
        return t
    else
        $E \leftarrow$ Preprocess and initialize set of shared edges between $t$ and $T|_R$
        $R' \leftarrow R$            // Initialize $R'$ by setting it equal to input $R$
        $t' \leftarrow t$              // Initialize $t'$ by setting it equal to input $t$
        for $x \in S \setminus R$ do
            $R' \leftarrow R' \cup \{x\}$
            $T' \leftarrow T|_{R'}$
            $t' \leftarrow$ AddLeaf($x$, $t'$, $T'$, $E$)
            $E \leftarrow$ Update shared edges between $t'$ and $T'$
        end
        return $t'$
    end
```

node that is incident to one of the edges in the backbone

Definition 3.3. There are exactly two types of superleaves, Type I and Type II:

1. A superleaf is a *Type I superleaf* if the edge e in the backbone to which the superleaf is attached is a shared edge in $T|_R$ and t. It follows then that a superleaf X is a Type I superleaf if and only if there exists a bipartition $A|B$ in $C(t) \cap C(T|_R)$ where $A|(B \cup X)$ and $(A \cup X)|B$ are both in $C(T|_{R \cup X})$.

2. A superleaf is a *Type II superleaf* if the edge e in the backbone to which the superleaf is attached is a unique edge in $T|_R$ and t. It follows that a superleaf X is a Type II superleaf if and only if for any bipartition $A|B$ such that $A|(B \cup X)$ and $(A \cup X)|B$ are both in $C(T|_{R \cup X})$, $A|B \notin C(t)$.

Now we begin our proof by establishing a lower bound on the RF distance to T for all binary S-completions of t.

Lemma 3.1. Let Y be a Type II superleaf for the pair (T, t), and let $x \in S \setminus R$. Let t^* be the result of adding x into t arbitrarily (i.e., we do not attempt to minimize the resulting RF distance). If $x \notin Y$, then Y is a Type II superleaf for the pair (T, t^*). Furthermore, if $x \in Y$, then $RF(T|_{R\cup\{x\}}, t^*) \geq RF(T|_R, t) + 2$.

Proof. It is easy to see that if $x \notin Y$, then Y remains a Type II superleaf after x is added to t. Now suppose $x \in Y$. We will show that we cannot add x into t without increasing the RF distance by at least 2. Since Y is a Type II superleaf, it is attached to a unique edge in $T|_{R\cup Y}$, and this is the same edge that x is attached to in $T|_{R\cup\{x\}}$. So suppose that x is added to t by subdividing an arbitrary edge e' in t with bipartition C|D; note that we do not require that x is added to a shared edge in t. After adding x to t we obtain tree t^* whose bipartition set includes $C|(D \cup \{x\})$ and $(C \cup \{x\})|D$. If C|D corresponds to a unique edge relative to t and $T|_R$, then both of these bipartitions correspond to unique edges relative to t^* and $T|_{R\cup\{x\}}$. If C|D corresponds to a shared edge, then at most one of the two new bipartitions can correspond to a shared edge, as otherwise we can derive that Y is a Type I superleaf. Hence, the number of unique edges in t must increase by at least one no matter how we add x to t, where x belongs to a Type II superleaf. Since t is binary, the tree that is created by adding x is binary, so that $RF(T|_{R\cup\{x\}}, t^*) \geq RF(T|_R, t) + 2$. QED.

Lemma 3.2. Let T^* be an unrooted binary tree that is a S-completion of t. Then we have $RF(T^*, T) \geq r + 2m$, where $r = RF(T|_R, t)$ and m is the number of Type II superleaves for the pair (T, t).

Proof. We note that adding a leaf can never reduce the total RF distance. The proof follows from Lemma 3.1 by induction. QED.

Now that we have established a lower bound on the best achievable RF distance (i.e., the optimality criterion for the RF Optimal Tree Completion problem), we show OCTAL outputs a tree T' that is guaranteed to achieve this lower bound. We begin by noting that when we add x to t by subdividing some edge e', creating a new tree t', all the edges other than e' in t continue to "exist" in t' although they define new bipartitions. In addition, e' is split into two edges, which can be considered new. Thus, we can consider whether edges that are shared between t and T *remain* shared after x is added to t.

Lemma 3.3. Let t' be the tree created by AddLeaf, given input tree t on leaf set R and tree T on leaf set $R \cup \{x\}$. If x is added to tree t by subdividing edge e' to create t', then all edges in t other than e' that are shared between t and T remain shared between t' and T.

Proof. Let $T^{(x)}$ be the rooted tree obtained by rooting T at x and then deleting x. Let e be the edge in $T^{(x)}$ corresponding to e', and let $\pi_e = A|B$; without loss of generality assume A is a clade in $T^{(x)}$. Note that $C(T)$ contains bipartition $A|(B \cup \{x\})$ (however, $C(T)$ may not contain $(A \cup \{x\})|B$, unless e is incident with the root of $T^{(x)}$). Furthermore, for subclade $A' \subseteq A$, $A'|(R \setminus A') \in C(T|_R)$ and $A'|(R \setminus A' \cup \{x\}) \in C(T)$. Now suppose e^* in t is a shared edge between t and $T|_R$ that defines bipartition $C|D \neq A|B$. Since $A|B$ and $C|D$ are both bipartitions of t, without loss of generality either $C \subset A$ or $A \subset C$. If $C \subset A$, then C is a clade in $T^{(x)}$, and so e^* defines bipartition $C|(D \cup \{x\})$ within t'. But since $C \subset A$, the previous analysis shows that $C|(D \cup \{x\})$ is also a bipartition of T, and so e^* is shared between T and t'. Alternatively, suppose $A \subset C$. Then within t', e^* defines bipartition $(C \cup \{x\})|D$, which also appears as a bipartition in T. Hence, e^* is also shared between T and t'. Therefore, any edge e^* other than e' that is shared between t and T remains shared between t' and T, for all leaves x added by AddLeaf. QED.

Lemma 3.4. OCTAL(T, t) preserves the topology of superleaves in T. In other words, for any superleaf with leaves $Q \subseteq S$, OCTAL(T, t)$|_Q$ equals $T|_Q$.

Proof. We will show this by induction on the number of leaves added. The lemma is trivially true for the base case when just one leaf is added to t. Let the inductive hypothesis be that the lemma holds for adding up to n leaves to t for some arbitrary $n \in \mathbb{N}^+$. Now consider adding $n + 1$ leaves, and choose an arbitrary subset of n leaves to add to t, creating an intermediate tree t' on leaf set K using the algorithm OCTAL. Let x be the next additional leaf to be added by OCTAL.

If x is the first element of a new superleaf to be added, it is trivially true that the topology of its superleaf is preserved, but we need to show that x will not break the monophyly of an existing superleaf in t'. By the inductive hypothesis, the topology of each superleaf already placed in t' has been preserved. Thus, each superleaf placed in t' has some shared edge in t' and $T|_K$ incident to that superleaf. If x were placed onto an edge contained in some existing superleaf, that edge would change its status from being shared to being unique, which contradicts Lemma 3.3.

The last case is where x is part of a superleaf for the pair (T, t) that already has been partially added to t. AddLeaf roots $T|_{K \cup \{x\}}$ at x and removes the edge incident to x, creating rooted tree $T^{(x)}$. The edge incident to the root in $T^{(x)}$ must be a shared edge by the inductive hypothesis. Thus, OCTAL adds x to this shared edge, preserving the superleaf topology. QED.

Lemma 3.5. OCTAL(T, t) returns binary tree T' such that $RF(T, T') = r + 2m$, where m is the number of Type II superleaves for the pair (T, t) and $r = RF(T|_R, t)$.

Proof. We will show this by induction on the number of leaves added.
Base Case: Assume $|S \setminus R| = 1$. Let x be the leaf in S$\setminus R$. AddLeaf adds x to a shared edge of t corresponding to some bipartition A|B, which also exists in $T^{(x)}$.

1. First we consider what happens to the RF distance on the edge x is attached to.

   If x is a Type I superleaf, the edge incident to the root in $T^{(x)}$ will be a shared edge by the definition of Type I superleaf, so AddLeaf adds x to the corresponding edge e' in t. The two new bipartitions that are created when subdividing e' will both exist in T by the definition of Type I superleaf so the RF distance does not change.

   If x is a Type II superleaf, either $(A \cup \{x\})|$B or $A|(B \cup \{x\})$ must not exist in $C(T)$. Since AddLeaf adds x to a shared edge, exactly one of those new bipartitions must exist in $C(T)$.

2. Now we consider what happens to the RF distance on the edges x is *not* attached to.

   Lemma 3.3 shows that AddLeaf (and therefore OCTAL) preserves existing shared edges between t and $T|_R$, possibly excluding the edge where x is added.

Thus, the RF distance will only increase by 2 if x is a Type II superleaf, as claimed.

Inductive Step: Let the inductive hypothesis be that the lemma holds for up to n leaves for some arbitrary $n \in \mathbb{N}^+$. Assume $|S \setminus R| = n + 1$. Now choose an arbitrary subset of leaves $Q \subseteq S \setminus R$, where $|Q| = n$, to add to t, creating an intermediate tree t' using the algorithm OCTAL. By the inductive hypothesis, assume t' is a binary tree with the RF distance between $T|_{Q \cup R}$ and t' equal to $r+2m$, where m is the number of Type II superleaves in Q. AddLeaf adds the remaining leaf $x \in S \setminus R$ to a shared edge of t' and $T|_{Q \cup R}$.

1. Lemma 3.3 shows that AddLeaf (and therefore OCTAL) preserves existing shared edges between t' and $T|_{Q \cup R}$, possibly excluding the edge where x is added.

2. Now we consider what happens to the RF distance on the edge x is attached to. There are three cases: (i) x is not the first element of a superleaf (ii) x is the first element of a Type I superleaf or (iii) x is the first element of a Type II superleaf.

   Case (i): If x is not the first element of a superleaf to be added to t, it directly follows from Lemma 3.4 that OCTAL will not change the RF distance when adding x.

   Case (ii): If x is the first element of a Type I superleaf to be added, then x is attached to a shared edge in the backbone corresponding to some bipartition $A|B$ existing in both $C(t)$ and $C(T|_R)$. Let e' be the edge in t s.t. $\pi_{e'} = A|B$. Note there must exist an edge e in $T|_{Q \cup R}$ producing $A|B$ when restricted to just R. Hence, the bipartition π_e has the form $M|N$ where $(M \cap R) = A$ and $(N \cap R) = B$. We need to show that $M|N \in C(t')$.

- By Lemma 3.3, any leaves from Q not attached to e' by OCTAL will preserve this shared edge in t'.
- Now consider when leaves from Q are added to e' by OCTAL. We decompose M and N into the subsets of leaves existing in either R or Q: let $M = A \cup W$ and $N = B \cup Z$. OCTAL will not cross a leaf from W with a leaf from Z along e' because this would require crossing the shared edge dividing these two groups: any leaf $w \in W$ has the property that $(A \cup \{w\})|B$ is a shared edge and any leaf $z \in Z$ has the property that $A|(B \cup \{z\})$ is a shared edge. Hence, any leaves added from Q that subdivide e' will always preserve an edge between leaves contained in W and Z on e'.

Thus, $M|N \in C(t')$. Moreover, $(M \cup \{x\})|N$ and $M|(N \cup \{x\})$ are bipartitions in $C(T)$. AddLeaf roots T at x and removes the edge incident to x, creating rooted tree $T^{(x)}$. We have shown that the edge incident to the root in $T^{(x)}$ must be a shared edge, so adding x does not change the RF distance.

Case (iii): If x is the first element of a Type II superleaf to be added, we have shown in Lemma 3.1 that the RF distance must increase by at least two. Since AddLeaf always attaches x to some shared edge e', the RF distance increases by exactly 2 when subdividing e'.

Thus, OCTAL will only increase the RF distance by 2 if x is a new Type II superleaf. QED.

Combining the above results, we establish our main theorem:

Theorem 3.1. Given unrooted binary trees t and T with the leaf set of t a subset of the leaf set of T, OCTAL(T, t) returns an unrooted binary tree T' that is a completion of t and that has the smallest possible RF distance to T. Hence, OCTAL finds an optimal solution to the RF Optimal Tree Completion problem. Furthermore, OCTAL runs in $O(n^2)$ time, where T has n leaves.

Proof. To prove that OCTAL solves the RF Optimal Tree Completion problem optimally, we need to establish that OCTAL returns an S-completion of the tree t, and that the RF distance between the output tree T' and the reference tree T is the minimum among all S-completions. Since OCTAL always returns a binary tree and only adds leaves into t, by design it produces a completion of t and so satisfies the first property. By Lemma 3.5, the tree T' output by OCTAL has an RF score that matches the lower bound established in Lemma 3.2. Hence, OCTAL returns a tree with the best possible score among all S-completions.

We now show that OCTAL can be implemented to run in $O(n^2)$ time, as follows. The algorithm has two stages: a preprocessing stage that can be completed in $O(n^2)$ time and a second stage that adds all the leaves from $S \setminus R$ into t that also takes $O(n^2)$ time.

In the preprocessing stage, we annotate the edges of T and t as either shared or unique, and we compute a set A of pairs of shared edges (one edge from each tree that define the same bipartition on R). We pick $r \in R$, and we root both t and T at r. We begin by computing, for each of these rooted trees, the LCA (least common ancestor) matrix for all pairs of nodes (leaves and internal vertices) and the number n_u of leaves below each node u; both can be computed easily in $O(n^2)$ time using dynamic programming. (For example, to calculate the LCA matrix, first calculate the set of leaves below each node using dynamic programing, and then calculate the LCA matrix in the second step using the set of leaves below each node.) The annotation of edges in t and T as shared or unique, and the calculation of the set A, can then be computed in $O(n^2)$ time as follows. Given an edge $e \in E(T)$, we note the bipartition defined by e as $X|Y$, where X is the set of leaves below e in the rooted version of T. We then let u denote the LCA of X in t, which we compute in $O(n)$ time (using $O(n)$ LCA queries of pairs of vertices, including internal nodes, each of which uses $O(1)$ time since we already have the LCA matrix). Once we identify u, we note the edge e' above u in t. It is easy to see that e is a shared edge if and only if e and e' induce the same bipartition on R, and furthermore this holds if and only if $n_u = |X|$. Hence, we can determine if e is a shared edge, and also its paired edge e' in t, in $O(n)$ time. Each edge in T is processed in $O(n)$ time, and hence the preprocessing stage can be completed in $O(n^2)$ time.

After the preprocessing, the second stage inserts the leaves from $S \setminus R$ into t using AddLeaf, and each time we add a leaf into t we have to update the set of edges of t (since it grows through the addition of the new leaf) and the set A. Recall that when we add $s \in S \setminus R$ into t, we begin by rooting T at s, and then follow a path towards the leaves until we find a first shared edge; this first shared edge may be the edge incident with s in T or may be some other edge, and we let e denote the first shared edge we find. We then use the set A to identify the edge $e' \in E(t)$ that is paired with e. We subdivide e' and make s adjacent to the newly created node. We then update A, the set of bipartitions for each tree, and the annotations of the edges of t and T as shared or unique. By Lemma 3.3, AddLeaf preserves all existing shared edges other than the edge the new leaf x is placed on, and these specific edges in E can each be updated in $O(1)$ time. Furthermore, OCTAL places x on a shared edge, bifurcating it to create two new edges. Thus, just two edges need to be checked for being shared, which again can be done in $O(n)$ as claimed. Thus, adding s to t and updating all the data structures can be completed in $O(n)$ time. Since there are at most n leaves to add, the second stage can be completed in $O(n^2)$ time. Hence, OCTAL runs in $O(n^2)$ time, since both stages take $O(n^2)$ time. QED.

3.4 EVALUATION AND RESULTS

3.4.1 Evaluation Overview

We compared OCTAL to the heuristic used in ASTRAL-II [70] for completing incomplete gene trees (see [129] for description), noting however that the ASTRAL-II technique is used to expand the search space explored by ASTRAL-II and does not explicitly attempt to minimize the distance to a reference tree. We used simulated datasets generated for [70] that have heterogeneity between gene trees and species trees due to ILS. To evaluate the accuracy of completed trees, we use three criteria: the normalized RF distance, normalized quartet distance, and the matching distance (see below for details).

We performed three sets of experiments:

- The first set of experiments evaluated the relative and absolute performance of ASTRAL-II and OCTAL for three levels of ILS (moderate, high, and very high) under these three evaluation criteria. The impact of the amount of missing data and gene tree estimation error was also examined.

- The second set of experiments evaluated the impact of the number of genes on the performance of ASTRAL-II and OCTAL. We restricted these experiments to two levels of ILS (moderate and high) and one evaluation criterion (normalized RF distance).

- The third set of experiments evaluated the impact of changing the reference tree on OCTAL. We again restricted these experiments to two levels of ILS (moderate and high) and one evaluation criterion (normalized RF distance).

3.4.2 Simulated datasets

The datasets used in this simulation study were originally generated for the ASTRAL-II study [70] and then modified for the purpose of this study. The full details of the protocol are described in [70], and briefly summarized here.

ASTRAL-II datasets. SimPhy [130] was used to simulate a collection of model species trees and, for each species tree, a collection of gene trees (with branch lengths deviating from a molecular clock) under the multi-species coalescent (MSC) model with varying levels of ILS. We refer to these simulated trees as the true gene trees and true species trees. Under this protocol, the true gene trees contain all the species, and the only cause for discordance between the true gene trees and the true species tree is ILS. For each individual true gene tree, INDELible [131] was used to simulate DNA sequences under the GTR+Γ model of evolution without insertions or deletions. The numeric model parameters varied across the gene trees and were determined by drawing from a distribution based on biological datasets. There are 50 replicate datasets per model condition.

Our modifications. We restricted the datasets examined in this study, by using only 26 species (one outgroup and 25 out of 200 ingroup taxa) and 200 out of 1000 genes. We examined 20 out of 50 replicate datasets for three model conditions: moderate ILS, high ILS, and very high ILS. We characterize the levels of ILS by the average normalized RF distance, referred to as "AD", between the true gene trees and the true species tree, calculated using Dendropy v4.2.0 [132]. Across all replicate datasets, the average AD was 10% for the moderate ILS condition, 36% for the high ILS condition, and 75% for the very high ILS condition.

We modified all datasets to ensure that some genes were incomplete, as follows. In each replicate (containing 200 genes), 150 genes were randomly selected to be missing data. In order to determine the number of taxa to be deleted from each gene, we noted the number of taxa in each non-trivial clade in the species tree; this produced a multi-set of numbers that vary between 2 and 20. Then for those genes that were selected to have taxa deleted, we selected a number n from the multi-set uniformly at random and selected n taxa to be deleted from the gene at random. This produced a set of 150 incomplete gene trees that on average were missing approximately 60% of the species. The estimated gene trees were computed using RAxML v8.2.8 [133] under the GTR+Γ model from the resulting alignments (i.e., all the sequences for the complete gene trees, and a subset of the sequences for the incomplete gene trees). This produced a set of 200 estimated gene trees (150 of which were incomplete) for every model condition and replicate dataset.

3.4.3 Gene tree completion

We used two techniques to complete the incomplete gene trees: the heuristic in ASTRAL-II and OCTAL. For the first set of experiments, ASTRID v1.4 was used to create reference trees for OCTAL. Both OCTAL and ASTRAL-II were run 9,000 times (150 incomplete gene trees in each of 20 replicates for three ILS levels).

As the amount of available data could potentially impact the quality of the reference tree used in OCTAL as well as the distance matrix computed by ASTRAL-II, we reduced the number of genes in the second set of experiments. In particular, we restricted the original 200-gene datasets to 25, 50, and 100 genes of which 5, 10, and 25 of these genes were complete, respectively; we also only explored the moderate and high ILS conditions, as these are closer to biological datasets. ASTRID v1.4 was again used to create reference trees for OCTAL, and both OCTAL and ASTRAL-II were run an additional 5,400 times.

Finally, in the third set of experiments, we directly evaluated the choice of reference tree on OCTAL by using the true species tree, the ASTRID v1.4 [71] tree, a greedy consensus tree, or a random tree drawn from a uniform distribution. Note that the ASTRID tree was computed on the full set of estimated gene trees (both incomplete and complete), while the greedy consensus tree was computed on the subset of estimated gene trees that were complete. For this final set of experiments, OCTAL was run an additional 18,000 times.

3.4.4 Evaluation criteria

We report error rates only for gene trees that were completed by ASTRAL-II or OCTAL, and we examined three different error metrics: normalized RF distance, normalized quartet distance, and matching distance. The normalized distances produce values that range from 0 to 1; all three distances return 0 only for those pairs of trees that are topologically identical, and so, low scores are better than large scores. The normalized RF distance between the completed estimated gene trees and the true gene trees was computed using Dendropy v4.2.0. This produces a value between 0 and 1, where 0 indicates that the completed estimated gene tree exactly matches the true gene tree, and 1 indicates that the two trees have no common bipartitions. The quartet distance between two trees on the same leaf set considers the quartet topologies induced by restricting each tree to all sets of four leaves (i.e. n choose four combinations, where n is the number of leaves). The quartet distance is then defined as the number of quartets that induce different topologies in the two trees. The matching distance between two trees on the same leaf set is the weight of a minimum weight perfect matching of their bipartitions, where each edge in the matching is weighted by the number of leaves that must be moved in order to transform one bipartition into its paired bipartition in the other tree [17].

We used one-sided paired Wilcoxon Signed-Rank tests [134] to determine whether using OCTAL (with the ASTRID tree) was significantly better than ASTRAL-II on each replicate dataset. As 20 replicate datasets were tested per model condition, a Bonferroni multiple comparison correction [135] was applied (i.e., p-values indicating significance must be less than 0.0025).

3.4.5 Commands

- Maximum likelihood gene trees were estimated using RAxML v8.2.8 (where input is the multiple sequence alignment for a given gene):
  ```
  raxmlHPC-SSE -m GTRGAMMA -p [seed] -n [name] -s [input]
  ```
- The random trees were created as follows. A star tree was created from the complete taxon set (i.e., the taxa in the complete trees). This star tree was then randomly resolved into a binary tree so that "the polytomy will be resolved by sequentially... generating all tree topologies equiprobably" [136]. Specifically, the random tree was generated using Dendropy v4.2.0:
  ```
  from dendropy.simulate import treesim
  from dendropy.utility import GLOBAL_RNG
  star_tree = treesim.star_tree(original_taxon_namespace)
  star_tree.resolve_polytomies(limit=2, rng=GLOBAL_RNG)
  ```
- The greedy consensus trees were computed using Bali-Phy v2.3.8 [137], where the input is the set of 50 complete RAxML trees (i.e., trees on the full taxon set):
  ```
  trees-consensus --greedy-consensus [input] [output]
  ```
- The command for ASTRID v1.4 (input is the full set of 200 RAxML trees):
  ```
  ASTRID-linux -i [input] -o [output]
  ```
- The command for ASTRAL v4.10.2 (input is the full set of 200 RAxML trees):
  ```
  java -jar astral.4.10.12.jar -i [input] -o [output]
  ```
- The normalized RF distances were computed using Dendropy v4.2.0:
  ```
  ne1 = len(tr1.internal_edges(exclude_seed_edge=True))
  ne2 = len(tr2.internal_edges(exclude_seed_edge=True))

  [fp, fn] = false_positives_and_negatives(tr1, tr2)
  rf = float(fp + fn) / (ne1 + ne2)
  ```
- The quartet distances were computed using QDist[138]:
  ```
  module load openblas/0.2.8-gcc
  module load gcc/6.2.0
  ./qdist tr1 tr2
  ```

- The matching distances were computed using code provided by the authors from [17], and now available at [139]:
  ```
  ./matching_distance tr1 tr2 numberofleaves
  ```

3.4.6 Experiment 1 Results: Performance of OCTAL and ASTRAL-II under 3 ILS levels

Results under moderate ILS levels. This experiment compared OCTAL (using ASTRID as the reference tree) to ASTRAL-II when given 200 genes (150 incomplete and 50 complete) under the moderate ILS level (AD=10%). The median RF error rate for ASTRAL-II was 17%, and the median RF error rate for OCTAL was 13% (Fig. 3.2). Using the RF error rate, OCTAL had better accuracy than ASTRAL-II on 1,366 genes, ASTRAL-II had better accuracy on 363 genes, and the methods were tied on the remaining 1,271 genes (Table 3.1). The degree of improvement in RF rate varied, but was as great as 20% on some datasets. The improvement obtained by using OCTAL over ASTRAL-II was statistically significant in 18 out of 20 of the replicates with this evaluation metric (Fig. 3.3).

Both the matching distance and quartet distance produced similar trends to the RF distance under the moderate ILS level. The median matching distance was 18 for ASTRAL-II and 15 for OCTAL (Fig. 3.2) and the improvement obtained by using OCTAL over ASTRAL-II was statistically significant in 19 out of 20 of the replicates. The median normalized quartet distance was 7% for ASTRAL-II and 6% for OCTAL (Fig. 3.2) and the improvement obtained by using OCTAL over ASTRAL-II was statistically significant in 18 out of 20 of the replicates.

The degrees of missing data and gene tree error did not impact whether OCTAL improved over ASTRAL-II under any of the evaluation metrics. We show our results for missing data with the RF error rate in Figure 3.4. Additional results for missing data with the matching distance and quartet distance show the same trend. Under very high levels of gene tree estimation error, there was a greater degree of improvement of OCTAL over ASTRAL-II with the RF error rate (Fig. 3.5). Additional results for gene tree error with the matching distance and quartet distance show a similar, though less pronounced, trend.

Results under high ILS. This experiment compared OCTAL (using ASTRID as the reference tree) to ASTRAL-II when given 200 genes (150 incomplete and 50 complete) under the high ILS level (AD=36%). OCTAL and ASTRAL-II achieved similar levels of accuracy under the high ILS condition, with both methods having a median RF error rate of 39% (Fig. 3.2). OCTAL was more accurate than ASTRAL-II on 1,004 genes, ASTRAL-II was more accurate on 524 genes, and the methods were tied on the remaining 1,472 genes

(Table 3.1). OCTAL provided a statistically significant advantage over ASTRAL-II in 7 of the 20 replicates, and the differences between the two methods were not statistically significant on the remaining 13 replicates (Fig. 3.3).

Again, the matching distance and quartet distance produced similar trends to the RF distance. The median matching distance was 41 for ASTRAL-II and 38 for OCTAL (Fig. 3.2), and the improvement obtained by using OCTAL over ASTRAL-II with respect to the matching distance was statistically significant in 10 out of 20 of the replicates. The median normalized quartet distance was 24% for ASTRAL-II and 23% for OCTAL (Fig. 3.2), and the improvement in quartet distance obtained by using OCTAL over ASTRAL-II was statistically significant in 5 out of 20 of the replicates.

Whether OCTAL or ASTRAL-II performed best appeared unrelated to the degree of missing data or gene tree estimation error under all evaluation criteria that we considered. The impact of missing data and the impact of gene tree estimation error on the RF error rate are shown in Figures 3.4 and 3.5, respectively. Matching distance and the quartet distance showed similar results.

Results under very high ILS. This experiment compared OCTAL (using ASTRID as the reference tree) to ASTRAL-II when given 200 genes (150 incomplete and 50 complete) under the very high ILS level (AD=75%). Using the RF error rate, OCTAL and ASTRAL-II achieved similar levels of accuracy, with both methods having a substantially increased median RF error rate of 78% (Fig. 3.2). OCTAL was more accurate than ASTRAL-II on 906 genes, ASTRAL-II was more accurate on 520 genes, and the methods were tied on the remaining 1,574 genes. OCTAL provided a statistically significant advantage over ASTRAL-II with the RF error rate in only 6 of the 20 replicates (Fig. 3.3).

In this case, the median matching distance was 77 for ASTRAL-II and 75 for OCTAL (Fig. 3.2), and the improvement obtained by using OCTAL over ASTRAL-II was statistically significant in 8 out of 20 of the replicates using the matching distance. The median normalized quartet distance was 51% for ASTRAL-II and 50% for OCTAL (Fig. 3.2) and the improvement in quartet distance obtained by using OCTAL over ASTRAL-II was statistically significant in 2 out of 20 of the replicates.

As we observed for the other ILS conditions, whether OCTAL or ASTRAL-II performed best appears unrelated to the degree of missing data or gene tree estimation error with respect to all evaluation criteria we considered. For the impact on RF error rate, Figure 3.4 shows results for missing data and Figure 3.5 shows results for gene tree error. The remaining results for the matching distance and the quartet distance showed a similar trend.

3.4.7 Experiment 2 Results: Impact of the number of genes on ASTRAL-II and OCTAL

As the number of genes determines the amount of data to be used in constructing a reference tree (required by OCTAL) and a distance matrix (required by ASTRAL-II), we varied the number of genes to see if this would impact the performance of OCTAL (using ASTRID as the reference tree) or ASTRAL-II under the moderate and high ILS conditions. Specifically, we examined subsets of the original 200-gene datasets with 25, 50, and 100 genes, of which 5, 10, and 25 were complete, respectively. As seen in Figure 3.6, under moderate ILS (AD=10%), ASTRAL-II had a median RF error rate of 22% (for 25 and 50 genes) and 17% (for 100 and 200 genes), whereas OCTAL had a median RF error rate of 17% (for 25, 50, and 100 genes) and 13% (for 200 genes). Hence, OCTAL was generally more accurate (as measured by the RF error rate) than ASTRAL-II under the moderate ILS condition. The relative improvement of OCTAL over ASTRAL-II per gene tree was 7% ± 4% (mean ± standard deviation) (i.e., 1-2 bipartitions) for all numbers of genes; however, the number of cases for which OCTAL improved over ASTRAL-II varied with the number of genes (see Table 3.2).

Results under high ILS (AD=36%) show somewhat different trends. ASTRAL-II had a median RF error rate of 48% for 25 genes, 44% for 50 genes, and 39% for 100 and 200 genes. OCTAL had lower median error rates at 25 (44% and 39%, respectively) but matched the median error rates of ASTRAL-II at 100 and 200 genes. However, OCTAL and ASTRAL-II have clearly different distributions for 200 genes (Figs. 3.2 and 3.6), so that even though the medians are the same OCTAL seems to provide a slight advantage over ASTRAL-II. Thus, on the high ILS datasets, OCTAL provided an improvement over ASTRAL-II, and the relative improvement per gene tree was similar to performance under the moderate ILS level (7-8% on average); however, there were fewer genes for which OCTAL improved over ASTRAL-II (see Table 3.2).

3.4.8 Experiment 3 Results: Impact of the reference tree on the performance of OCTAL

Our final experiment examined the impact of reference tree on OCTAL on the 200-gene datasets with moderate and high levels of ILS, using the RF error rate as the evaluation criterion. We considered four reference trees: 1) the true species tree, 2) the ASTRID species tree computed on the all gene trees (50 complete and 150 incomplete), 3) the greedy consensus tree computed on the 50 complete gene trees, and 4) a random tree on the same set of species. The greedy consensus tree, also known as the extended majority consensus tree, is obtained by ordering the bipartitions from the input set of trees according to their frequency, and then adding them one-by one-to a growing set of bipartitions if they are compatible with the set.

The ASTRID and greedy consensus trees had low species tree RF error (at most 9% RF) under the moderate ILS condition and somewhat higher species tree error (at most 22% RF) when the level of ILS was high. We found that there was little difference (less than 1% in median gene tree RF error) between using ASTRID, a greedy consensus of the complete gene trees, and even the true species tree, as the reference tree (Fig. 3.7). However, using a random tree as the reference tree produced extremely high RF error rates for the completed trees, which is as expected as the random species tree had extremely high error: between 96% and 100% RF for each replicate.

3.5 DISCUSSION

OCTAL is a greedy polynomial time algorithm that adds species into an estimated gene tree so as to provably minimize the RF distance to a given reference tree. In our study, OCTAL frequently produced more accurate completed gene trees than ASTRAL-II under ILS conditions ranging from moderate to very high; however, the improvement under high ILS conditions was much lower and less frequent than under moderate ILS conditions. This trend does not appear to be sensitive to the distance measure used to evaluate the accuracy of the completed gene trees. We offer the following as a hypothesis for the reason for this trend. Under low to moderate ILS, the true species tree is close to the true gene tree, and the estimated species trees (computed using ASTRID or the greedy consensus) are reasonably close to the true species tree; by the triangle inequality, the estimated species tree is close to the true gene trees. Therefore, when ILS is at most moderate, completing the estimated gene trees using the estimated species tree as a reference can be beneficial. However, under higher ILS, the true species tree is farther from the true gene trees, which makes the true species tree (or an estimate of that tree) less valuable as a reference tree.

We also saw that using estimated species trees as reference trees produced comparably accurate completions as using the true species tree as a reference, and that this held for both moderate and high ILS levels. In particular, using several techniques to produce the reference tree from the gene trees, including even a greedy consensus tree, produced reference trees that were as good as the true species tree in terms of the impact on the accuracy of the completed gene tree. However, a random tree produced very poor results. Hence, OCTAL was robust to moderate levels of error in the estimated species tree. However, OCTAL is not completely agnostic to the choice of reference tree, since the random reference tree (which has close to 100% RF error) resulted in very poor performance. We note that since the completion of this work, an algorithm [53] with a better running time was developed for solving the RF-OTC introduced here.

There are many directions for future work. First, we compared OCTAL to ASTRAL-II, but ASTRAL-III [140] has recently been developed, and the comparison should be made to this new version of ASTRAL. OCTAL could also be compared to gene tree completion methods that are designed to handle gene tree heterogeneity resulting from gene duplication and loss [141], and these comparisons could be made on datasets that have evolved under multiple causes of gene tree discord (e.g., GDL, HGT, and ILS).

The optimal completion problem or the accuracy of the completed gene trees could also be based on other distances between trees besides the RF distance, including weighted versions [142] of the RF distance (where the weights reflect branch lengths or bootstrap support values), quartet tree distances, geodesic distances [143], or the matching distance. It is likely that some of these problems will be NP-hard, but approximation algorithms or heuristics may be useful in practice.

We did not evaluate the impact of using OCTAL on downstream analyses. Since missing data (i.e., incomplete gene trees) are known to impact species tree estimation methods using summary methods [128], this would be a natural next analysis. As an example, if the input includes some incomplete gene trees, a species tree could be estimated from the full set of gene trees, and OCTAL could use that estimated species tree as a reference tree to complete the gene trees. Then, the species tree could be re-estimated using a summary method on the new set of gene trees, all of which are complete. This two-step process (completing gene trees using an estimated species tree, then re-estimating the species tree) could then iterate. It would be interesting to determine whether this improves the species tree, and if so under what conditions. It would also be helpful to evaluate the impact of completing incomplete gene trees when the genes are missing due to true biological loss rather than data collection issues, and hence also to see if OCTAL provides any helpful insight into gene evolution (such as better estimating the duplication/loss/transfer parameters).

Finally, there can be multiple optima to the RF Optimal Tree Completion problem for any given pair of trees, and exploring that set of optimal trees could be important. An interesting theoretical question is whether the set of optimal solutions admits a compact representation, even when it is large. From a practical perspective, the set of optimal completions could be used to provide support values for the locations of the missing taxa, and these support values could then be used in downstream analyses.

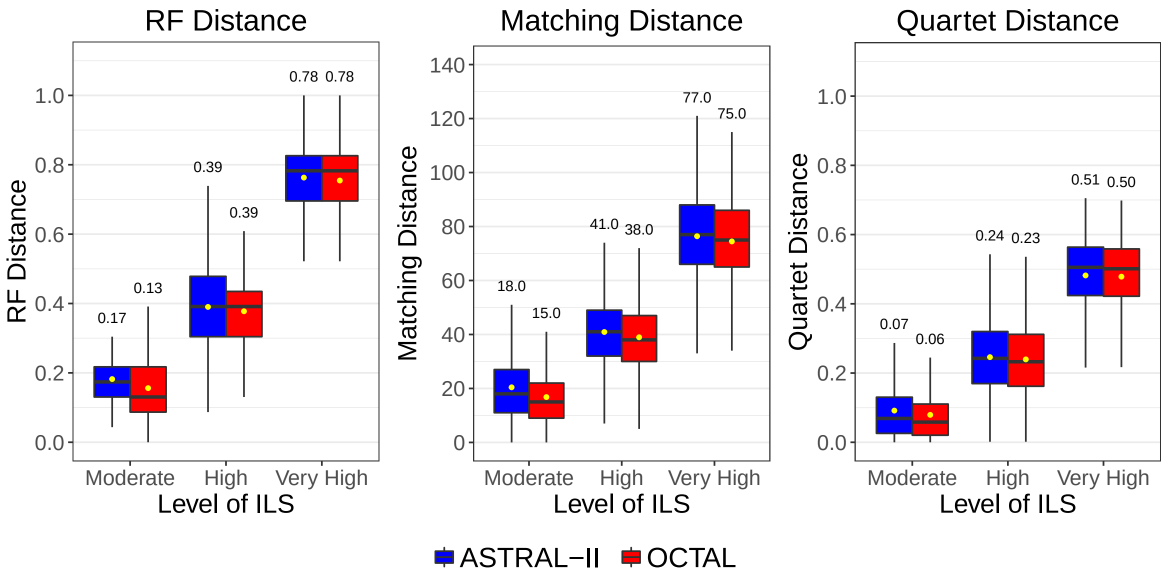


Figure 3.2: **The performance of OCTAL and ASTRAL-II across three levels of ILS evaluated under three tree distance metrics.** Each subfigure shows the performance of OCTAL in red (using ASTRID as the reference tree) and ASTRAL-II in blue under one of three distance metrics. Under each distance metric, a lower value indicates the estimated completed tree is closer to the true gene tree. The median distance is reported above each boxplot, and so the outliers are not shown. OCTAL shows the largest improvement over ASTRAL-II under the moderate ILS condition in each case.

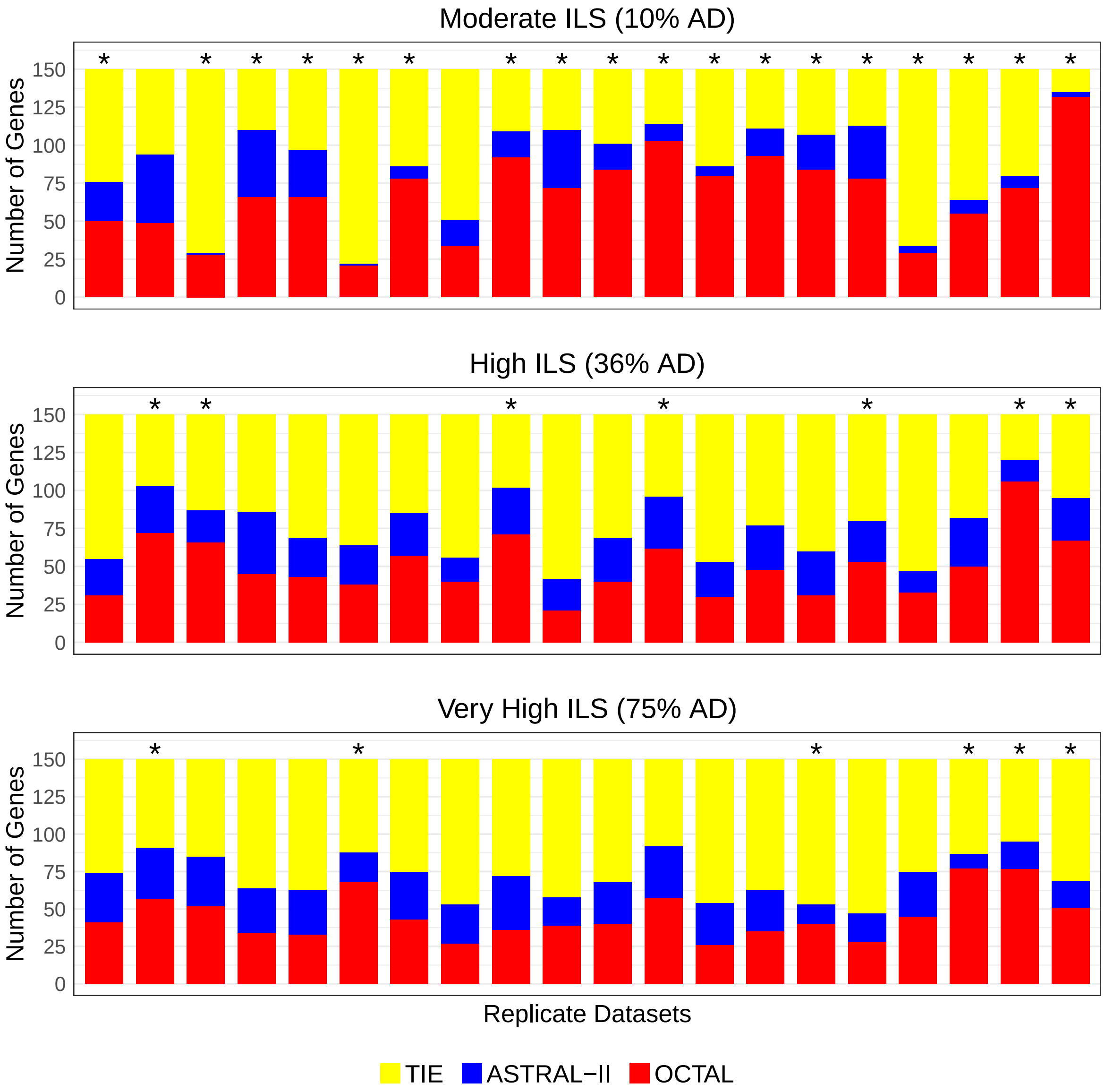


Figure 3.3: **The performance of OCTAL and ASTRAL-II across replicate datasets with the RF distance evaluation criteria.** Each subfigure shows the relative performance of OCTAL (using ASTRID as the reference tree) and ASTRAL-II where RF distance was used to compare the estimated completed gene trees to the true gene trees. The number of gene trees for which OCTAL is better than ASTRAL-II is shown in red, the number of gene trees for which ASTRAL-II is better is shown in blue, and the number of genes for which OCTAL and ASTRAL-II are tied is shown in yellow. OCTAL has a statistically significant improvement over ASTRAL-II (as measured by a one-sided Wilcoxon signed-rank test; see main text for details) on replicate datasets with an asterisk (*).

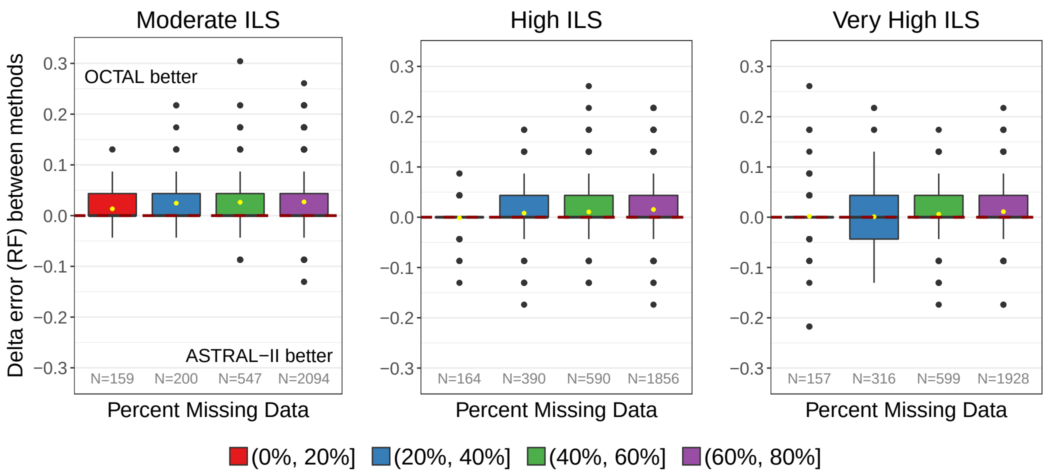


Figure 3.4: **The impact of degree of missing data on relative performance of OCTAL and ASTRAL-II under the RF distance evaluation criteria.** The y-axis shows the difference in the RF error rate between trees completed using OCTAL (using ASTRID as the reference tree) and ASTRAL-II. Positive values indicate that OCTAL is better than ASTRAL-II, and negative values indicate that ASTRAL-II is better. For many genes, there is no difference in accuracy between OCTAL and ASTRAL-II. However, when there is a difference between the two methods, OCTAL frequently outperforms ASTRAL-II. This finding holds regardless of the degree of missing data. For each level of ILS, boxplots include genes with a specified percent of missing data (e.g., red indicates genes are missing 0-20% of the species). The number N of genes in each plot is provided on the x-axis.

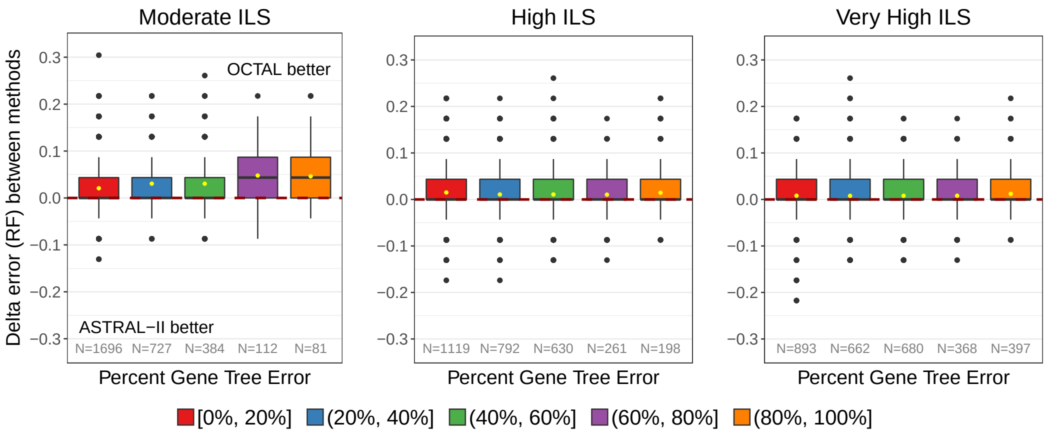


Figure 3.5: **The impact of gene tree estimation error on relative performance of OCTAL and ASTRAL-II under the RF distance evaluation criteria.** The y-axis shows the difference in the RF error rate between trees completed using OCTAL (using ASTRID as the reference tree) and ASTRAL-II. Positive values indicate that OCTAL is better than ASTRAL-II, and negative values indicate that ASTRAL-II is better. For each level of ILS, boxplots include genes with the specified percent of gene tree estimation error. The number N of genes in each plot is provided on the x-axis.

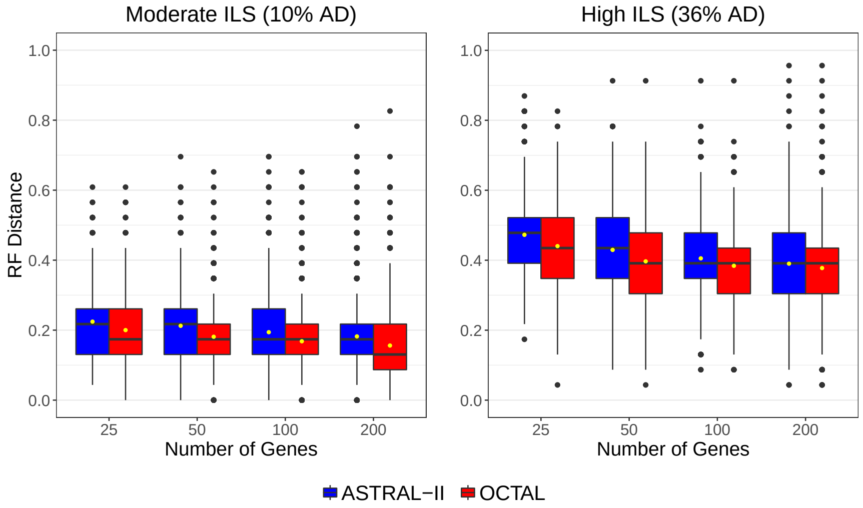


Figure 3.6: **The performance of OCTAL and ASTRAL-II for varying numbers of genes under the RF distance evaluation criteria.** The x-axis shows the number of genes varying from 25 to 200. The y-axis shows the RF error rate between the true gene trees and the gene trees completed using OCTAL with the ASTRID reference tree (red) or ASTRAL-II (blue). The number of data points per boxplot varies with the number of genes. For example, the 25-genes model condition has 400 data points per boxplot (20 incomplete genes across 20 replicates), whereas the 200-gene model condition has 3,000 data points per boxplot (150 incomplete genes across 20 replicates).

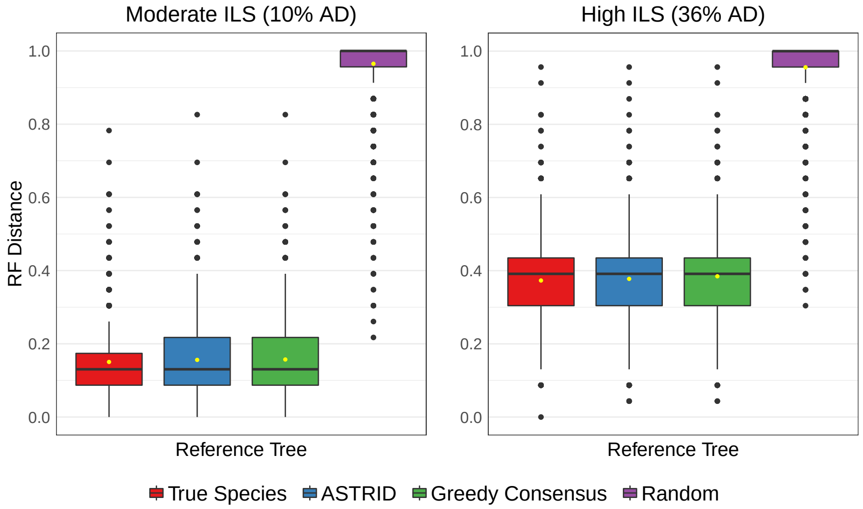


Figure 3.7: **Impact of reference tree on OCTAL with the RF distance evaluation metric.** The x-axis shows the reference tree used by OCTAL. The y-axis shows the RF error rate between the true gene trees and the gene trees computing using OCTAL (varying the reference tree). Only the 200-gene model condition is shown, so each boxplot has 3,000 data points (150 incomplete genes across 20 replicates).

Table 3.1: The number of gene trees for which OCTAL provided an improvement over ASTRAL-II, for which ASTRAL-II provided an improvement of OCTAL, and for which there was no difference between the two methods is provided below for three levels of ILS and three evaluation distance criteria. The RF, matching, and quartet distances are used for evaluating the distance between the completed, estimated trees and the true gene trees. Numbers in bold indicate the largest number of genes. OCTAL improves more genes than ASTRAL-II except in the higher ILS conditions with the RF distance criteria, in which case OCTAL and ASTRAL-II are more often equal in their performance.

Error Metric	OCTAL better	ASTRAL-II better	No difference
Moderate ILS (AD=10%)			
RF	**1366**	363	1271
Matching	**1666**	522	812
Quartet	**1540**	594	866
High ILS (AD=35%)			
RF	1004	524	**1472**
Matching	**1501**	920	579
Quartet	**1473**	1092	435
Very high ILS (AD=75%)			
RF	906	520	**1574**
Matching	**1643**	1143	214
Quartet	**1552**	1371	77

Table 3.2: The number of gene trees for which OCTAL provided an improvement over ASTRAL-II, for which ASTRAL-II provided an improvement of OCTAL, and for which there was no difference between the two methods is provided below for two levels of ILS and four numbers of genes. The RF error rate is used for evaluating the distance between the completed, estimated trees and the true gene trees. Numbers in bold indicate the largest number of genes. OCTAL improves more genes than ASTRAL-II except when the level of ILS is high and the number of genes is 200, in which case OCTAL and ASTRAL-II are more often equal in their performance.

Number of genes	OCTAL better	ASTRAL-II better	No difference
Moderate ILS (AD=10%)			
25 genes	**177**	62	161
50 genes	**420**	116	262
100 genes	**685**	188	627
200 genes	**1366**	363	1271
High ILS (AD=35%)			
25 genes	**228**	79	93
50 genes	**398**	119	283
100 genes	**624**	265	611
200 genes	1004	524	**1472**

CHAPTER 4: CORRECTING UNRESOLVED BRANCHES IN GENE TREES USING A REFERENCE TREE WITH TRACTION

In the previous chapter, we looked at how a reference tree could be used to improve gene trees afflicted with missing data. Here, we build upon this approach by using a reference tree to not only add missing species, but also correct gene trees with low-support branches. Figures and tables appear at the end of this chapter in Section 4.6.

4.1 INTRODUCTION

While species tree estimation techniques can exploit information encoded across the entire genome, gene tree estimation based on a single locus may not contain enough signal to determine the correct gene tree topology with high confidence [51]. Indeed, many phylogenomic datasets have gene trees with average branch support well below 75%, which is a common lower bound for branches to be considered reliable. For example, the Avian Phylogenomic Project [146] reported average branch support values below 30%, and many other studies have had similar challenges (surveyed in [128]). Estimating gene and species trees is further complicated by biological processes such as GDL, ILS, and HGT, which create heterogeneous tree topologies across the genome [10] and prevent the naive adoption of one tree topology.

In cases where estimated gene trees have high uncertainty, the work in the previous chapter suggests that we may be able to improve these gene trees using a reference tree. Approaches from the GDL literature have previously worked on closely related problems as they are interested in gene and species tree reconciliation for the purposes of identifying orthologs, dating duplication events, and finding sites of adaptation [147]. It was from this line of work that improving uncertain branches in gene trees to be more consistent with the species tree then emerged organically under GDL-aware models of evolution. As introduced in Chapter 2, gene tree correction methods [54, 55, 61], which only rely on the topologies of gene and species trees, as well as integrative methods [51, 52, 53], which also incorporate sequence data, have been successful at correcting and reconciling gene trees. So far, the methods that have been developed are based on parametric models of gene evolution under

This chapter contains material previously presented at the 2019 *Workshop on Algorithms in Bioinformatics* [144] and later published in *Algorithms for Molecular Biology* under the title "Non-parametric correction of estimated gene trees using TRACTION" [145]. This work was done in conjunction with E.K. Molloy, P. Vachaspati, A. Yammanuru, and T. Warnow; TW conceived the project; SC, PV designed and implemented the algorithm; SC, EKM, PV, AY, TW established theory; SC, AY, TW wrote proofs; SC, PV, EKM performed experiments; SC, EKM, AY produced the figures; SC, EKM, PV, AY, TW contributed to the writing of this chapter.

GDL. These parameterized models may not generalize and have not been tested in other contexts in which gene and species tree discord is driven by other biological phenomena.

Here, we examine gene tree correction where gene tree heterogeneity is due to ILS or HGT, and where each gene tree has at most one copy of each species. We present a new approach to gene tree correction that is based on a very simple *non-parametric* polynomial-time method, the Tree Refinement And CompleTION (TRACTION) algorithm. In addition to correcting gene trees, TRACTION is also capable of completing gene trees that do not contain all the species present in the reference species tree, a condition that may occur in a multi-locus study when not all genomes have been sequenced and assembled.

The input to TRACTION is a pair (t, T) of unrooted, singly-labeled phylogenetic trees. The leaf set of t is a subset of the leaf set of T, tree T is binary, and tree t will generally be non-binary. We seek a tree T' created by refining t and adding any missing leaves so that T' has the minimum Robinson-Foulds (RF) [11] distance to T (see Def. 2.1). We call this the *RF-Optimal Tree Refinement and Completion Problem* (RF-OTRC) and show that TRACTION finds an optimal solution to RF-OTRC in $O(n^{1.5} \log n)$ time, where n is the number of leaves in the species tree T. We also explore an extension of this problem statement to handle multi-labeled genes by using a generalization of the RF distance proposed in [148].

To use TRACTION for gene tree correction in practice, we assume we are given an estimated gene tree with branch support values and an estimated (or known) binary species tree, which may have additional species. The low support branches in the gene tree are collapsed, forming the (unresolved) tree t. TRACTION first refines the input gene tree t into a binary tree t', and then it adds the missing species to t' using OCTAL from Chapter 3. Although the algorithm is quite simple, the proof of correctness is non-trivial.

We present the results of an extensive simulation study on 68,000 gene trees in which gene tree heterogeneity is either due to only ILS or to both ILS and HGT. We explore TRACTION for gene tree correction with estimated species trees in comparison to GDL correction methods, including Notung [54, 55], ecceTERA [61], ProfileNJ [51], TreeFix [52], and TreeFix-DTL [53]. Many methods, including TRACTION, tie for best on the ILS-only data, but TRACTION dominates the other gene tree correction methods with respect to topological accuracy on the HGT+ILS data, while also tying for fastest. Importantly, TRACTION provides good accuracy even when the estimated species tree is far from the true gene tree. The simplicity of the approach and its good accuracy under a range of model conditions indicate that non-parametric approaches to gene tree correction may be promising and encourages future research.

4.2 PROBLEM STATEMENT

We now turn our attention to the optimization problem of interest to this paper. This section is limited to the context of singly-labeled trees, and we hold off the extension to MUL-trees until a later section.

Problem 4.1 (Optimal Tree Refinement and Completion (RF-OTRC))**.** Given an unrooted, singly-labeled, binary tree T on leaf set S and an unrooted, singly-labeled tree t on $R \subseteq S$ output an unrooted, singly-labeled, binary tree T' on S with two key properties: (1) T' contains all the leaves of S and is compatible with t (i.e., $T'|_R$ is a refinement of t), and (2) T' minimizes the RF distance to T among all binary trees satisfying condition (1).

If the trees t and T have the same set of taxa, then the RF-OTRC problem becomes the RF-OTR (RF-Optimal Tree Refinement) problem. If t is already binary but can be missing taxa, then the RF-OTRC problem becomes the RF-OTC (RF-Optimal Tree Completion) problem (introduced above in Problem 3.1). OCTAL [118], presented in Algorithm 3.1, solves the RF-OTC problem in $O(n^2)$ time, and an improved approach presented by Bansal [149] solves the RF-OTC problem in linear time. We refer to this faster approach as *Bansal's algorithm*. In this paper we present an algorithm that solves the RF-OTR problem exactly in polynomial time and show that the combination of this algorithm with Bansal's algorithm solves the RF-OTRC problem exactly in $O(n^{1.5} \log n)$ time, where T has n leaves. We refer to the two steps together as TRACTION.

4.3 METHODS

The input to TRACTION is a pair of unrooted, singly-labeled trees (t, T), where t is the estimated gene tree on set R of species and T is the binary reference tree on S, with $R \subseteq S$. We note that t may not be binary (e.g., if low support edges have already been collapsed) and may be missing species (i.e., $R \subset S$ is possible).

Step 1: Refine t to produce a binary tree t^* maximizing shared bipartitions with T.

Step 2: Add the missing species from T into t^*, minimizing the RF distance.

Step 1: Greedy Refinement of t. To compute t^*, we first refine t by adding all bipartitions from $T|_R$ that are compatible with t; this produces a unique tree t'. If t' is not fully resolved, then there are multiple optimal solutions to the RF-OTR problem, as we will later prove. The algorithm selects one of these optimal solution as follows. First, we add edges from t that were previously collapsed (if such edges are available). Next, we randomly refine the tree until we obtain a fully resolved refinement, t^*. Note that if t' is not binary, then t^* is not unique. We now show that the first step of TRACTION solves the RF-OTR problem.

Theorem 4.1. Let T be an unrooted, singly-labeled tree on leaf set S, and let t be an unrooted, singly-labeled tree on $R \subseteq S$. A fully resolved (i.e. binary) refinement of t minimizes the RF distance to $T|_R$ if and only if it includes all compatible bipartitions from $T|_R$.

Proof. Let C_0 denote the set of bipartitions in $T|_R$ that are compatible with t. By the theoretical properties of compatible bipartitions, this means the set $C_0 \cup C(t)$ is a compatible set of bipartitions that define a unique tree t' where $C(t') = C_0 \cup C(t)$ (since the trees are singly-labeled). We now prove that for any binary tree B refining t, B minimizes the RF distance to $T|_R$ if and only if B refines t'.

Consider a sequence of trees $t = t_0, t_1, t_2, \ldots, t_k$, each on leaf set R, where t_i is obtained from t_{i-1} by adding one edge to t_{i-1}, and thus adds one bipartition to $C(t_{i-1})$. Let $\delta_i = RF(t_i, T|_R) - RF(t_{i-1}, T|_R)$, so that δ_i indicates the change in RF distance produced by adding a specific edge to t_{i-1} to get t_i. Hence,

$$RF(t_i, T|_R) = RF(t_0, T|_R) + \sum_{j \leq i} \delta_j. \tag{4.1}$$

A new bipartition π_i added to $C(t_{i-1})$ is in $C(T|_R)$ if and only if $\pi_i \in C_0$. If this is the case, then the RF distance will decrease by one (i.e., $\delta_i = -1$). Otherwise, $\pi_i \notin C_0$, and the RF distance to $T|_R$ will increase by one (i.e., $\delta_i = 1$).

Now suppose B is a binary refinement of t. We can write the bipartitions in $C(B) \setminus C(t)$ into two sets, X and Y, where X are bipartitions in C_0 and Y are bipartitions not in C_0. By the argument just provided, it follows that $RF(B, T|_R) = RF(t, T|_R) - |X| + |Y|$. Note that $|X \cup Y|$ must be the same for all binary refinements of t, because all binary refinements of t have the same number of edges. Thus, $RF(B, T|_R)$ is minimized when $|X|$ is maximized, so B minimizes the RF distance to $T|_R$ if and only if $C(B)$ contains all the bipartitions in C_0. In other words, $RF(B, T|_R)$ is minimized if and only if B refines t'. QED.

Corollary 4.1. TRACTION finds an optimal solution to the RF-OTR problem.

Proof. Given input gene tree t and reference tree T on the same leaf set, TRACTION produces a tree t'' that refines t and contains every bipartition in T compatible with t; hence by Theorem 4.1, TRACTION solves the RF-OTR problem. QED.

Step 2: Adding in missing species. The second step of TRACTION can be performed using OCTAL or Bansal's algorithm, each of which finds an optimal solution to the RF-OTC problem in polynomial time. Indeed, we show that any method that optimally solves the RF-OTC problem can be used as an intermediate step to solve the RF-OTRC problem.

To prove this, we first restate several prior theoretical results [118]. Lemma 3.2 and Lemma 3.5 together imply that the minimum achievable RF distance between T and T' is given by:

$$RF(T, T') = RF(T|_R, t) + 2m \tag{4.2}$$

where m is the number of Type II superleaves in T relative to t, which we defined above (see Definitions 3.2 and 3.3). Furthermore, recall that Theorem 3.1, which states that given unrooted, singly-labeled binary trees t and T with the leaf set of t a subset of the leaf set S of T, OCTAL(T, t) solves the RF-OTC problem and runs in $O(n^2)$ time, where T has n leaves.

4.3.1 Proof of correctness for TRACTION

Lemma 4.1. Let T be an unrooted, singly-labeled, binary tree on leaf set S with $|S| = n$, and let t be an unrooted, singly-labeled tree on leaf set $R \subseteq S$. TRACTION returns a binary unrooted tree T' on leaf set S such that $RF(T', T)$ is minimized subject to $T'|_R$ refining t.

Proof. By construction *TRACTION* outputs a tree T' that, when restricted to the leaf set of t, is a refinement of t. Hence, it is clear that $T'|_R$ refines t. Now, it is only necessary to prove that RF(T', T) is minimized by *TRACTION*. Since the intermediate tree t^* produced in the first step of TRACTION is binary, Theorem 3.1 gives that TRACTION using OCTAL (or any method exactly solving the RF-OTC problem) will add leaves to t^* in such a way as to minimize the RF distance to T; hence it suffices to show that t^* computed by TRACTION has the smallest RF distance to T among all binary refinements of t.

As given in Equation 4.2, the optimal RF distance between T' and T is the sum of two terms: (1) RF(t^*, $T|_R$) and (2) the number of Type II superleaves in T relative to t^*. Theorem 4.1 shows that TRACTION produces a refinement t^* that minimizes the first term. All that remains to be shown is that t^* is a binary refinement of t minimizing the number of Type II superleaves in T relative to t^*.

Consider a superleaf X in T with respect to t. If t were already binary, then every superleaf X is either a Type I or a Type II superleaf. Also, note that every Type I superleaf in T with respect to t will be a Type I superleaf for any refinement of t. However, when t is not binary, it is possible for a superleaf X in T to be a Type II superleaf with respect to t but a Type I superleaf with respect to a refinement of t. This happens when the refinement of t introduces a new shared edge with T to which the superleaf X is attached in T. Notice that since the set of all possible shared edges that could be created by refining t is compatible, any refinement that maximizes the number of shared edges with T also minimizes the number of Type II superleaves. Theorem 4.1 shows that *TRACTION* produces such a refinement t^* of t. Thus, *TRACTION* finds a binary unrooted tree T' on leaf set S such that RF(T', T) is minimized subject to the requirement that $T'|_R$ refine t. QED.

Theorem 4.2. TRACTION solves the RF-OTRC problem and runs in $O(n^{1.5} \log n)$ time if used with Bansal's algorithm and $O(n^2)$ time if used with OCTAL, where n is the number of leaves in the input.

Proof. The above lemma shows that TRACTION solves the RF-OTRC problem. Let t, T, S, and R be as defined in the RF-OTRC problem statement. What remains to be shown is a running time analysis for the first stage of TRACTION (refining t). We claim this step takes $O(|S| + |R|^{1.5} \log(|R|))$ time.

Constructing $T|_R$ takes $O(|S|)$ time. Checking compatibility of a single bipartition with a tree on K leaves, and then adding the bipartition to the tree if compatible, can be performed in only $O(|K|^{0.5} \log(|K|))$ after a fast preprocessing step (see Lemmas 3 and 4 from [150]). Hence, determining the set of edges of $T|_R$ that are compatible with t takes only $O(|S| + |R|^{1.5} \log(|R|))$ time. Therefore, the first stage of TRACTION takes $O(|S| + |R|^{1.5} \log(|R|))$ time. Hence, if used with OCTAL, TRACTION takes $O(|S|^2)$ time and if used with Bansal's algorithm TRACTION takes $O(|S|^{1.5} \log |S|)$ time. QED.

4.3.2 Extending TRACTION to MUL-trees

Up to this point, we have formulated gene tree correction problems only in the context where the input trees are each singly-labeled (i.e., have at most one leaf for each species). However, in the context of GDL, a gene tree may have multiple copies of a species at its leaves (i.e., it can be a "MUL-tree"). We now generalize the RF-OTR problem to allow the input unresolved tree t to be a MUL-tree (although we still require the species tree to be singly-labeled). Note, we require both input trees to be on the same leaf set (i.e., we do not discuss a completion step) because, in the context of GDL, if a species is missing in the gene tree, this can be attributed to true biological loss.

The RF distance between two MUL-trees is defined to be the minimum number of contractions and refinements that suffice to transform one tree into the other, but computing this distance is NP-hard [148]. Also, the RF distance no longer translates to the number of bipartitions present in one tree but not the other (and in particular, two MUL-trees can have identical sets of bipartitions but not be isomorphic, as shown in [151]). To define the multi-labeled analog of RF distance so that it corresponds to bipartition distances, we introduce definitions and results from [151].

We represent a MUL-tree by a triplet $\mathcal{T} = (T, S, \phi)$ where T is an unlabeled tree, S is a set of labels, and ϕ is a surjective function mapping the leaves of T onto S. Note that in the special case where ϕ is a bijection, then the MUL-tree is a singly-labeled phylogenetic tree.

Now let $\mathcal{T}_1 = (T_1, S, \phi_1)$ be a MUL-tree. Then, a *full differentiation* of the MUL-tree $\mathcal{T}_1$ is a singly-labeled tree $\mathcal{T}_1^* = (T_1, S^*, \phi_1^*)$ where ϕ_1^* is a bijection (Definition 7 in [151]). More plainly, $\mathcal{T}_1^*$ has the same topology as $\mathcal{T}_1$, but has unique leaf labels. Let $\mathcal{T}_2 = (T_2, S, \phi_2)$ be another MUL-tree on set S with some full differentiation $\mathcal{T}_2^* = (T_2, S^*, \phi_2^*)$. $\mathcal{T}_1^*$ and $\mathcal{T}_2^*$ are *consistent* full differentiations if, for each label $\ell \in S$, the set of labels assigned to leaves in $\mathcal{T}_1^*$ that were labeled ℓ in $\mathcal{T}_1$ is identical to the set of labels assigned to leaves in $\mathcal{T}_2^*$ that were labeled ℓ in $\mathcal{T}_2$ (Definition 8 in [151]). Note that this requires $\mathcal{T}_1$ and $\mathcal{T}_2$ to have the same number of copies of ℓ at the leaves. Theorem 3 from [151] establishes that the RF distance between two MUL-trees $\mathcal{T}_1$ and $\mathcal{T}_2$ is equal to the minimum RF distance between any *mutually consistent full differentiations* (MCFDs) of $\mathcal{T}_1$ and $\mathcal{T}_2$ into singly-labeled trees:

$$RF(\mathcal{T}_1, \mathcal{T}_2) = \min\{RF(T_1, T_2) : T_1 \text{ and } T_2 \text{ are MCFDs of } \mathcal{T}_1 \text{ and } \mathcal{T}_2\text{, resp.}\} \tag{4.3}$$

We continue by introducing a generalization to RF distance proposed in [148] that allows us to calculate a distance between a MUL-tree and a singly-labeled tree. Let T be a singly-labeled tree on leaf set S and let $\mathcal{R} = (t, S, \phi)$ be a MUL-tree on that same leaf set. The *extension* of T with respect to $\mathcal{R}$ is the MUL-tree constructed from T by replacing each leaf s in T by an internal node connecting to k leaves labeled with s, where k is the number of copies of s in $\mathcal{R}$. In [148], the authors define the RF distance between $\mathcal{R}$ and T to be the RF distance between $\mathcal{R}$ and the extension of T. Note that this is now just the RF distance between two MUL-trees with the same number of copies for each leaf, which is well-defined under the classic RF definition. We now state the refinement problem generalized to MUL-trees, using this definition of RF distance.

Problem 4.2 (RF Optimal Tree Refinement Problem for Mul-Trees)**.** Given a MUL-tree $\mathcal{R} = (t, S, \phi)$ and an unrooted, binary singly-labeled tree T where $S = L(T)$, output an unrooted binary tree $\mathcal{R}'$ with two key properties: (1) $\mathcal{R}'$ refines $\mathcal{R}$, and (2) $\mathcal{R}'$ minimizes the RF distance to T among all binary trees satisfying condition (1).

In general, computing the RF distance between two MUL-trees is NP-hard, but Theorem

3 in [148] showed that the RF distance between a MUL-tree and an extended singly-labeled tree is equal for all MCFDs. Thus, one can pick any MCFD of the MUL-tree and the extended singly-labeled tree and apply the usual bipartition-based method for calculating the RF distance in polynomial time.

Theorem 4.3. Let MUL-tree $\mathcal{R} = (t, S, \phi)$ and singly-labeled tree T be an arbitrary instance of the RF-OTR problem for MUL-trees. Let $\mathcal{T}$ be the extension of T with respect to $\mathcal{R}$. Then TRACTION run on any MCFD of $\mathcal{R}$ and $\mathcal{T}$ solves this RF-OTR problem exactly.

Proof. Any MCFD of the input MUL-tree and the extended singly-labeled tree result in two singly-labeled trees, t^* and T^* respectively, on the same leaf set. By Theorem 4.1, Traction finds a refinement of t^*, which we call t', that minimizes the RF distance to T^*. We can naturally map t' back to a MUL-tree $\mathcal{R}' = (t', S, \phi')$ refining $\mathcal{R}$ by inverting the full differentiation of the leaves. Note that this step of TRACTION and this proof of solving the refinement problem optimally did not rely on the assumption that the reference tree is binary (in general, T^* is not binary).

Using Theorem 3 from [148], $\mathcal{R}'$ minimizes the RF distance to the extended singly-labeled tree $\mathcal{T}$ out of all refinements of $\mathcal{R}$. By the generalized definition of RF distance, $\mathcal{R}'$ then also minimizes the RF distance to the singly-labeled tree T out of all refinements of $\mathcal{R}$, proving the claim. QED.

Figure 4.1 provides example of a MUL-tree, an extended species tree, and TRACTION's solution to the RF-OTR problem for MUL-trees.

4.4 EVALUATION AND RESULTS

4.4.1 TRACTION under Gene Duplication and Loss: Case Study

There are model conditions under which TRACTION will not accurately modify an input estimated gene tree, even when given the true species tree as the reference tree and a collapsed version of the true gene tree. For example, if a duplication event takes place at the root of a species tree, then genes of the same species will not be siblings in the true gene tree. Hence, if TRACTION is given the true gene tree (i.e., MUL-tree), it will not be able to add any bipartitions to it from the extended species tree, and will instead return a random refinement (see Figure 4.2 a-c). For a second example, if a duplication event takes place closer to the leaves, then genes of the same species appear somewhat close to each other in the true gene tree. As a result, TRACTION may add edges in the wrong place, resulting in incorrect locations for duplications (see Figure 4.2 d-g). The key point to both cases is that when TRACTION adds edges from the extended species tree, these imply duplications at the leaves of the species tree, and the edges produced by random refinements of the MUL-tree have low probability (i.e., never more than $\frac{1}{3}$) of being in the true species tree.

4.4.2 TRACTION under ILS and HGT: Simulations

Overview. We evaluated TRACTION in comparison to two gene tree correction methods, Notung [54, 55] and ecceTERA [61], as well as three integrative methods, profileNJ [51], TreeFix [52], and TreeFix-DTL [53]. We tested each method on estimated gene trees under two different model conditions (ILS-only and ILS+HGT), using estimated and true species trees. In total, we analyzed 68,000 genes: 8,000 with 26 species under ILS-only models and 60,000 with 51 species under ILS+HGT models. All estimated gene trees that we correct in these experiments were complete (i.e., not missing species). The motivation for this is two-fold. First, the methods we benchmarked against do not provide an option for completing gene trees with missing data. This is understandable since these methods were developed for GDL, where missing species in a gene tree are interpreted as true loss events rather than incomplete sampling. Second, an experimental evaluation of OCTAL, the algorithm that performs the completion step of TRACTION, was previously performed in [118].

Datasets. All datasets used in this study are from prior studies [118, 152] and available online. The datasets included singly-labeled genes with 26 or 51 species (each with a known outgroup), and were generated under model conditions where true gene trees and true species trees differed due to only ILS (datasets with 26 species had two levels of ILS) or due to both ILS and HGT (datasets with 51 species had the same level of ILS but two different levels of HGT). The true gene tree heterogeneity (*GT-HET*, the topological distance between true species trees and true gene trees) ranged from 10% (for the ILS-only condition with moderate ILS) to as high as 68% (for the ILS+HGT condition with high HGT). Each model condition has 200 genes, and we explored multiple replicate datasets per model condition with different sequence lengths per gene. See Table 4.1 for details.

Estimated gene trees and estimated reference species trees. For each gene, we used RAxML v8.2.11 [133] under the GTRGAMMA model to produce maximum likelihood gene trees, with branch support computed using bootstrapping. Because sequence lengths varied, this produced estimated gene trees with different levels of gene tree estimation error *(GTEE)* (defined to be the average RF distance between the true gene tree and the estimated gene tree), ranging from 32% to 63% as defined by the missing branch rate (see Table 4.1). We estimated a species tree using ASTRID v1.4 [71] given the RAxML gene trees as input. Because the true outgroup for all species trees and gene trees was known, we rooted the species tree and all gene trees at the outgroup prior to performing gene tree correction.

The gene trees given as input to the different correction methods were computed as follows. Each gene tree estimated by RAxML had branches annotated with its bootstrap support, and we identified all the branches with bootstrap support less than a given threshold. These branches with low support were then collapsed in the gene trees before being given to TRACTION, Notung, and ProfileNJ. When we ran ecceTERA, we gave the binary gene trees with the threshold value (i.e., minimum required bootstrap support value); ecceTERA collapses all branches that have support less than the threshold value, and explores the set of refinements. Thus, the protocol we followed ensured that ecceTERA, ProfileNJ, Notung, and TRACTION all used the same set of collapsed gene trees. TreeFix and Treefix-DTL used the uncollapsed gene trees. We ran all methods using a threshold value of 75% (the standard threshold for low support). We additionally ran TRACTION and Notung using collapse thresholds of 50%, 85%, and 90% on the ILS-only data.

Gene tree correction and integrative methods. The RAxML gene trees were corrected using TRACTION v1.0, Notung v2.9, ecceTERA v1.2.4, ProfileNJ (as retrieved from GitHub after the March 20, 2018 commit with ID 560b8b2) [51], TreeFix v1.1.10 (for the ILS-only datasets), and TreeFix-DTL v1.0.2 (for the HGT+ILS datasets), each with a species tree estimated using ASTRID v1.4 [71] as the reference tree rooted at the outgroup. The integrative methods (TreeFix, TreeFix-DTL, and ProfileNJ) also required additional input data related to the gene alignments, which we detail in the commands below. All estimated gene trees were complete (i.e., there were no missing taxa), so TRACTION only refined the estimated gene tree and did not add any taxa. We also explored using the true model species tree as a reference tree for TRACTION and Notung on the ILS-only datasets.

Evaluation criteria. We used RF tree error, a standard criterion in performance studies, to quantify error in estimated and corrected gene trees as compared to the known true gene tree. Although we used RF distance within the OTR optimization criterion, in that context, it refers to the distance between the corrected gene tree and the reference tree (which is an *estimated species tree*); by contrast, the RF error rate in the evaluation criterion refers to the distance between the corrected gene tree and the true gene tree. Since the reference trees used in our experiments are typically very topologically different from the true gene tree (8% RF distance for the moderate ILS condition, 33% for the high ILS condition, 54% to 68%

for the ILS+HGT conditions, see Table 4.1), optimizing the RF distance to the reference tree is quite different from optimizing the RF distance to the true gene tree. Nevertheless, we also evaluated the methods using the matching [17] and quartet distance [138].

Experiments. We performed two main experiments: one in which we explored performance on ILS-only datasets and the other in which we explored performance on datasets with HGT and ILS. In each case, we directly explored how the GTEE level impacted absolute and relative accuracy of gene tree correction methods. We also indirectly explored how GT-HET affects relative and absolute accuracy. Heterogeneity is higher on the HGT+ILS datasets than on the ILS-only datasets, as HGT adds heterogeneity between gene trees and species trees (see Table 4.1). In our third experiment, we evaluated how the branch support collapse threshold and how using the true species tree as the reference tree impacted absolute and relative performance among the best performing methods on the ILS-only datasets.

Commands. In the following commands, *resolved gene trees* refers to the gene trees estimated using RAxML, *unresolved gene trees* refers to these estimated gene trees with branches having bootstrap support less than the threshold (e.g., 75%) collapsed, and *reference species tree* refers to the species tree estimated using ASTRID. *Rooted* means the input tree was rooted at the outgroup.

RAxML v8.2.11 was run as

```
raxml -f a -m GTRGAMMA -p 12345 -x 12345 -N <# bootstrap replicates> \
      -s <alignment file> -n <output name>
```

ASTRID v1.4 was run as

```
ASTRID -i <resolved gene trees> -o <output>
```

Notung v2.9 was run as

```
java -jar Notung-2.9.jar --resolve -s <rooted reference species tree> \
     -g <rooted unresolved gene tree> --speciestag postfix \
     --treeoutput newick --nolosses
```

TRACTION v1.0 was run as

```
traction.py --refine -r -s 12345 -b <unrooted reference species tree> \
         -u <unrooted resolved gene trees> \
         -i <unrooted unresolved gene trees> -o <output>
```

ecceTERA v1.2.4 was run as

```
eccetera resolve.trees=0 collapse.mode=1 collapse.threshold=75 \
         dated=0 print.newick=true \
         species.file=<rooted reference species tree> \
         gene.file=<rooted resolved gene tree>
```

FastME v2.1.6.1 [153], used to compute a distance matrix for ProfileNJ, was run as

```
fastme -i <input gene alignment> -O <output distance matrix> -dK
```

ProfileNJ, using the K2P-corrected distance matrix from FastME, was run as

```
profileNJ \
   -g  <rooted unresolved gene tree> -s  <rooted reference species tree> \
   -d  <distance matrix> -o  <output> -S  <name map> -r none \
   -c nj  --slimit 1 --plimit 1 --firstbest --cost 1 0.99999
```

TreeFix v1.1.10 was run on the ILS-only datasets as

```
treefix -s <rooted reference species tree> -S <name map> \
        -A <alignment file extension>  -o <old tree file extension> \
        -n <new tree file extension> <resolved gene tree>
```

TreeFix-DTL v1.0.2 was run on the HGT+ILS datasets as

```
treefixDTL -s <rooted reference species tree> -S <map file> \
           -A <alignment file extension> \
           -o <old gene tree file extension> \
           -n <new gene tree file extension> <resolved gene tree>
```

Normalized RF distances were computed using Dendropy v4.2.0 [132] as

```
n1 = len(t1.internal_edges(exclude_seed_edge=True))
n2 = len(t2.internal_edges(exclude_seed_edge=True))
[fp, fn] = false_positives_and_negatives(t1, t2)
rf = float(fp + fn) / (n1 + n2)
```

Matching distances were computed using code from [17] and [139] as

```
matching_distance <tree 1> <tree 2> <number of leaves>
```

Quartet distances were computed using QDist [138] as

```
qdist <tree 1> <tree 2>
```

4.4.3 Experimental Results

Experiment 1: Comparison of methods on ILS-only datasets. Not all methods completed on all datasets: ecceTERA failed to complete on 67 gene trees, profileNJ failed to complete on two gene trees, and all other methods completed on all gene trees. Results shown in Figure 4.3 are restricted to those datasets on which all methods completed. For the moderate ILS condition with accuracy evaluated using RF distance (Figure 4.3 (top)), all methods were able to improve on RAxML, and the degree of improvement increased with GTEE. For the high ILS condition (Figure 4.3 (bottom)), methods improved on RAxML only when GTEE was at least 20%. Thus, GTEE and ILS level both impacted whether methods improved on RAxML. Furthermore, the methods grouped into two sets: TRACTION, Notung, and TreeFix performing very similarly and ProfileNJ and ecceTERA having somewhat higher error. We found the relative performance of these methods follows the same trends for matching (Figure 4.4) and quartet distances (Figure 4.5) as for RF distances.

Experiment 2: Comparison of methods on the HGT+ILS datasets. The HGT+ILS datasets have heterogeneity due to both HGT and ILS, with the degree of HGT varying from moderate (m5) to high (m6). Here, ecceTERA failed on 1,318 datasets with the failure rates increasing as the GTEE of the initial RAxML gene tree increased: ecceTERA failed 0% of the time when GTEE was less than 40%, 0.4% of the time when GTEE was 40-60%, 23.6% of the time when GTEE was 60-80%, and 90.8% of the time when GTEE was at least 80%. Because of the high failure rate, we report results for ecceTERA on datasets with GTEE of at most 40%; above this level, ecceTERA fails frequently, making comparisons between methods potentially biased. Figure 4.6 shows that ecceTERA performed well, though not as well as Notung and TRACTION, on these low GTEE datasets.

Figure 4.7 shows the impact of the remaining methods on RAxML gene trees as a function of GTEE as measured by RF distance. Figure 4.8 and Figure 4.9 measure this impact using matching distance and quartet distance, respectively. The relative performance between the remaining methods across all evaluation metrics show that TRACTION and Notung were more accurate than profileNJ and TreeFix-DTL, with the gap between the two groups increasing with GTEE. We also see that TRACTION had an advantage over Notung for the low GTEE condition and matched the accuracy on the higher GTEE condition. For the lowest GTEE bin, no method improved the RAxML gene tree, but some methods made the gene trees less accurate (e.g., profileNJ); only TRACTION maintained the accuracy of the RAxML gene tree. Overall, on the HGT+ILS datasets, TRACTION consistently performed well and provided a clear advantage over the other methods in terms of accuracy.

Experiment 3: Varying collapse threshold and reference tree on the ILS datasets. The collapse threshold is an important hyperparameter that may impact the accuracy of gene tree correction methods. We evaluated the effect of this parameter on the two best performing methods from the previous experiments: TRACTION and Notung. Figure 4.10 shows the results on the ILS-only datasets, stratified by GTEE. Overall, TRACTION and Notung exhibited similar relative performance. Intuitively, increasing the collapse threshold (i.e., collapsing more branches) tends to reduce the error in the moderate ILS condition across all levels of GTEE as well the high ILS condition with sufficiently high GTEE. However, a lower threshold (i.e., collapsing fewer braches) improves accuracy for the low GTEE and high ILS condition, where the original gene tree is well-estimated and the reference species tree is more distant from the true gene trees.

The reference tree is also an important input that in practice will often itself be estimated. In Figure 4.11, we found that using the true model species tree achieves similar absolute performance as using the estimated ASTRID tree as reference. Again, TRACTION and Notung had performed similarly with respect to the RF distance between the true and the estimated (and then corrected) gene tree.

Running Times. We selected a random sample of the 51-taxon HGT+ILS datasets to evaluate the running time (see Table 4.2). From fastest to slowest, the average running times were 0.5 seconds for TRACTION, 0.8 seconds for Notung, 1.7 seconds for ProfileNJ, 3.8 seconds for TreeFix-DTL, and 29 seconds for ecceTERA. Most of the methods had consistent running times from one gene to another, but ecceTERA had high variability, depending on the size of the largest polytomy. When the largest polytomy was relatively small, it completed in just a few seconds, but it took close to a minute when the largest polytomy had a size at the limit of 12. Results on other HGT+ILS replicates and model conditions gave very similar results.

Overall comments. This simulation study shows that the better methods for gene tree correction (TRACTION, Notung, and TreeFix) produced more accurate gene trees than the initial RAxML gene trees for the ILS-only conditions (except for cases where the initial gene tree was already very accurate), and that the improvement could be very large when the initial gene trees were poorly estimated. However, the impact of gene tree correction was reduced for the HGT+ILS scenarios, where improvement over the initial gene tree was only obtained when GTEE is fairly high. As shown in Table 4.1, the average normalized RF distance between the reference tree (ASTRID) and the true gene trees was never more than 33% for the ILS-only scenarios but very high for the HGT+ILS scenarios (54% for moderate

HGT and 68% for high HGT). Since a reference tree (i.e., an estimated species tree) was the basis for the correction of the gene trees, it is not surprising that improvements in accuracy were difficult to obtain for the HGT+ILS scenario. On the other hand, given the large distance between the true species tree and the true gene tree, the fact that improvements were obtained for several methods (TRACTION, Notung, and TreeFix-DTL) is encouraging.

4.5 DISCUSSION

We presented TRACTION, a method that solves the RF-OTRC problem exactly in $O(n^{1.5} \log n)$ time, where n is the number of species in the species tree. TRACTION performs well on singly-labeled gene trees, matching or improving on the accuracy of competing methods on the ILS-only datasets and dominating the other methods on the HGT+ILS datasets. Although all the methods are reasonably fast on these datasets, TRACTION is the fastest on the 51-taxon gene trees with Notung a close second.

The observation that TRACTION performs as well (or better) than the competing methods (ecceTERA, ProfileNJ, Notung, TreeFix, and TreeFix-DTL) on singly-labeled gene trees under ILS and HGT is encouraging. However, the competing methods are all based on stochastic models of gene evolution that are inherently derived from GDL scenarios (and in one case also allowing for HGT), and thus it is not surprising that GDL-based methods do not provide the best accuracy on the ILS-only or HGT+ILS model conditions we explore. Yet, TRACTION has good accuracy under a wide range of scenarios for singly-labeled gene trees. We conjecture that this generally good performance is the result of its non-parametric criterion which can help it to be robust to model misspecification.

This study suggests several other directions for future research. The GDL-based methods have variants that may enable them to provide better accuracy (e.g., alternative techniques for rooting the gene trees, selecting duplication/loss parameter values, etc.), and future work should explore these variants. Most gene tree correction methods have been developed specifically to address the case where genes have multiple copies of species as a result of gene duplication events. We showed that a naive extension of TRACTION to handle multi-labeled genes by using a generalization of the RF distance based on an extended species tree, such as proposed in [148], can lead to misleading results. Future work should explore other generalizations of RF distance that do not suffer from these same limitations (e.g., [154]). Recent work has shown how Notung could be extended to address HGT [155]; a comparison between TRACTION and a new version of Notung that addresses HGT will need to be made when Notung is modified to handle HGT. Finally, the effect of gene tree correction on downstream analyses should also be evaluated carefully.

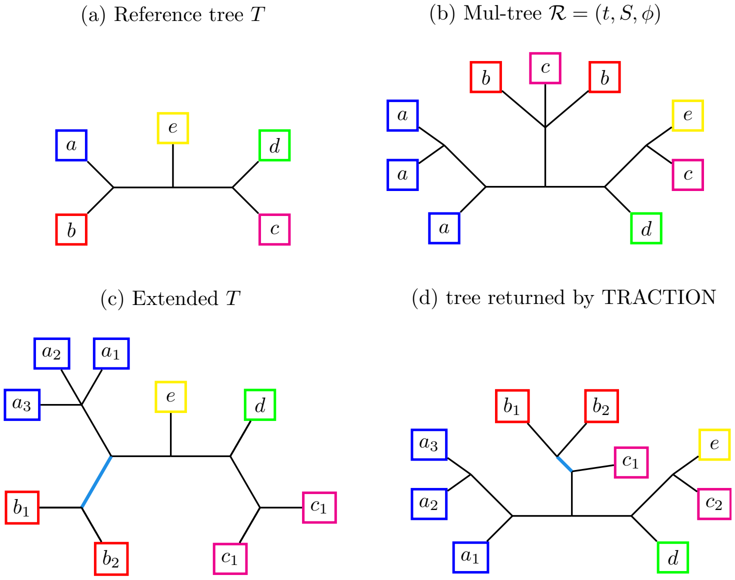
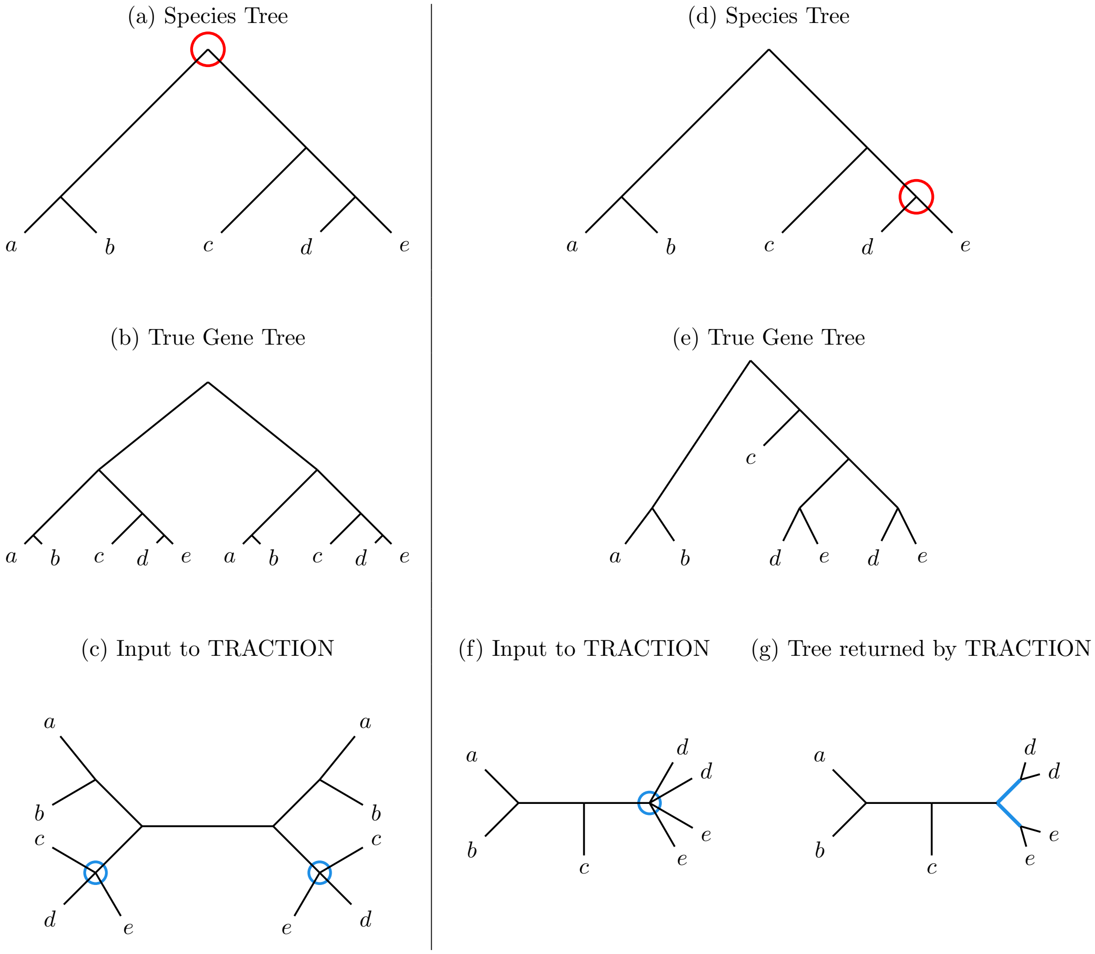


Figure 4.1: **Example of MUL-tree correction using TRACTION given a reference tree.** Given a singly-labeled, binary tree T on leaf set S, we wish to correct MUL-tree $\mathcal{T} = (t, S, \phi)$ using TRACTION. First, we build the extension of T with respect to t, called *Extended* T. We then re-label leaves such that t and T are MCFD and add any bipartitions from Extended T missing from t; the only such bipartition in this example is in blue.



(a) Species Tree

(b) True Gene Tree

(c) Input to TRACTION

(d) Species Tree

(e) True Gene Tree

(f) Input to TRACTION

(g) Tree returned by TRACTION

Figure 4.2: **Two cases where TRACTION does not perform well on multi-labeled gene trees.** In the first case (left column), a duplication event (red circle) occurs at the root of the species tree shown in (a), producing the true gene tree shown in (b). If TRACTION is given the estimated gene tree shown in (c) and the unrooted true species tree (a) as input, then TRACTION will randomly refine the estimated gene tree, because it cannot add any branches. In the second case (right column), a duplication event (red circle) occurs towards the leaves of the species tree shown in (d), producing the true gene tree shown in (e). If TRACTION is given the estimated gene tree shown in (f) and the unrooted true species tree (d) as input, then TRACTION will add two branches as shown in blue in (g), producing an incorrect gene tree. Furthermore, the addition of these two incorrect branches would imply two duplication events, one occurring at leaf *d* and one occurring at leaf *e*, in the true species tree, so that the gene tree returned by TRACTION will not minimize the number of duplication events.

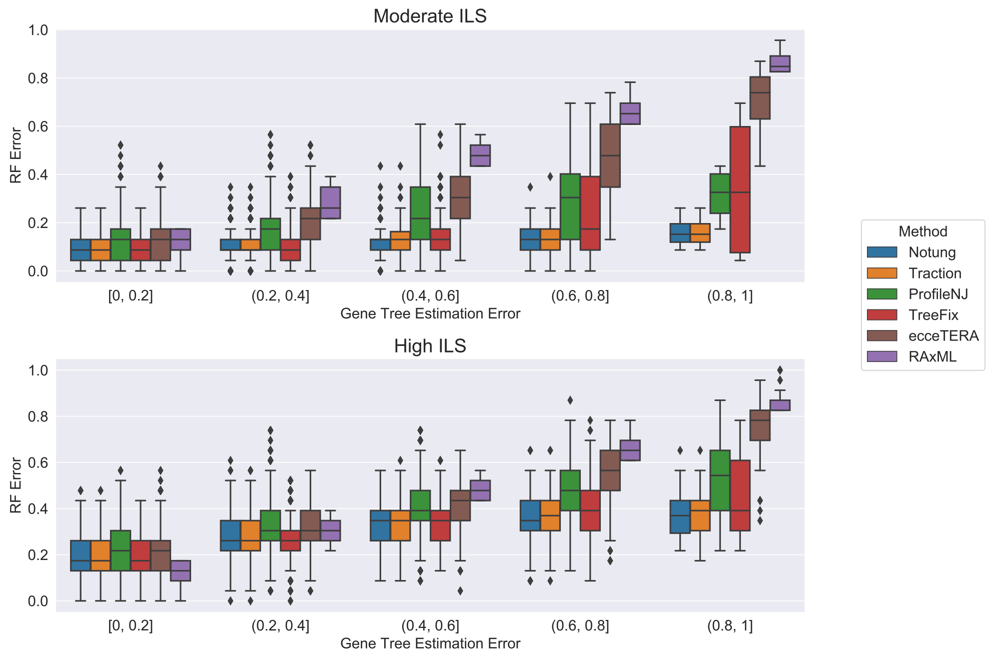


Figure 4.3: **Comparison of methods on the ILS-only datasets with respect to RF distance as a function of GTEE.** Results are only shown for those datasets on which all methods completed. Each model condition (characterized by ILS level) has 20 replicate datasets, each with 200 genes.

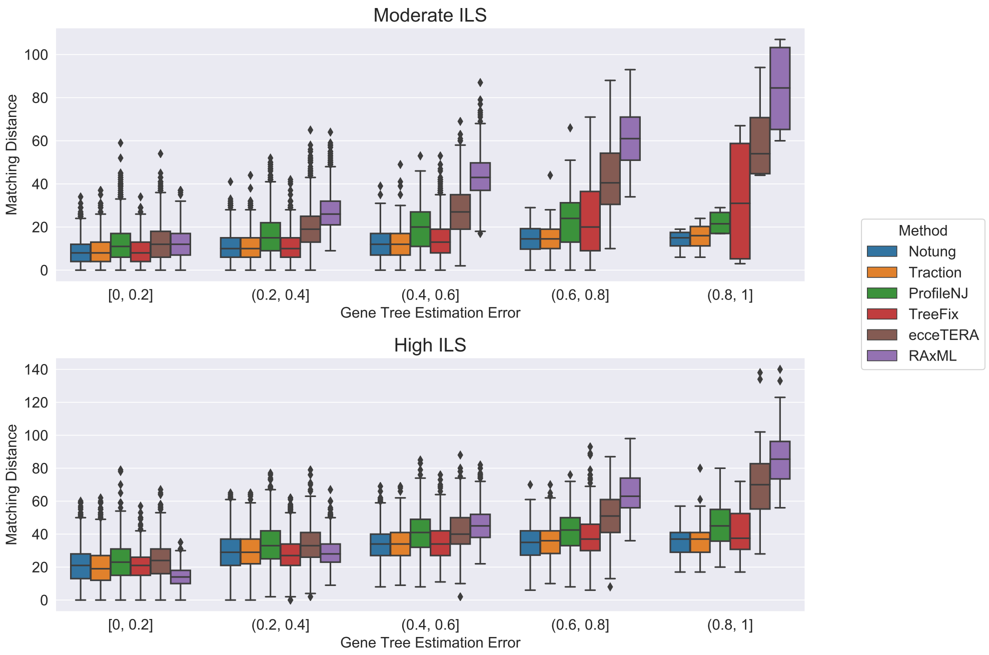


Figure 4.4: **Comparison of methods on the ILS-only datasets with respect to matching distance as a function of GTEE.** Results are only shown for those datasets on which all methods completed. Each model condition (characterized by ILS level) has 20 replicate datasets, each with 200 genes.

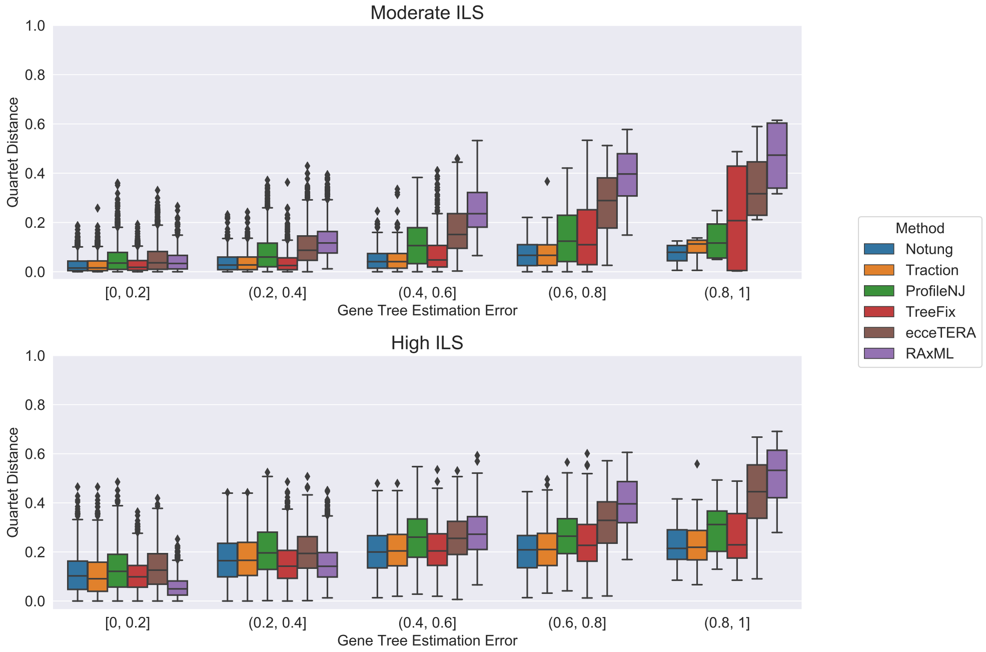


Figure 4.5: **Comparison of methods on the ILS-only datasets with respect to quartet distance as a function of GTEE.** Results are only shown for those datasets on which all methods completed. Each model condition (characterized by ILS level) has 20 replicate datasets, each with 200 genes.

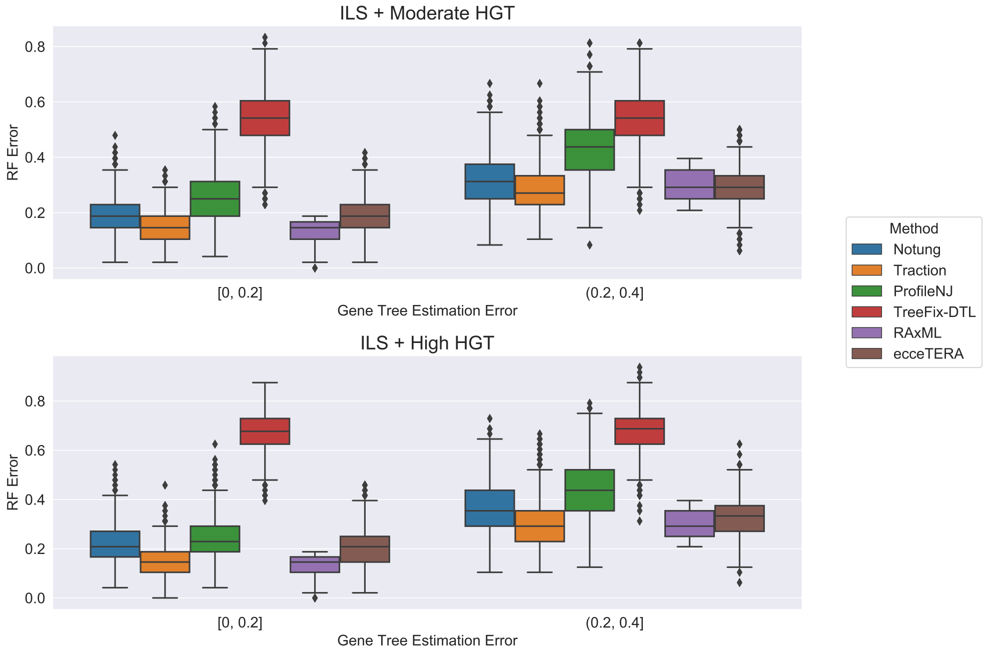


Figure 4.6: **ecceTERA performs relatively well on datasets on which it completes.** Boxplots show a comparison on ILS+HGT datasets with respect to RF error as a function of GTEE. We only show GTEE conditions for which ecceTERA completed on all genes.

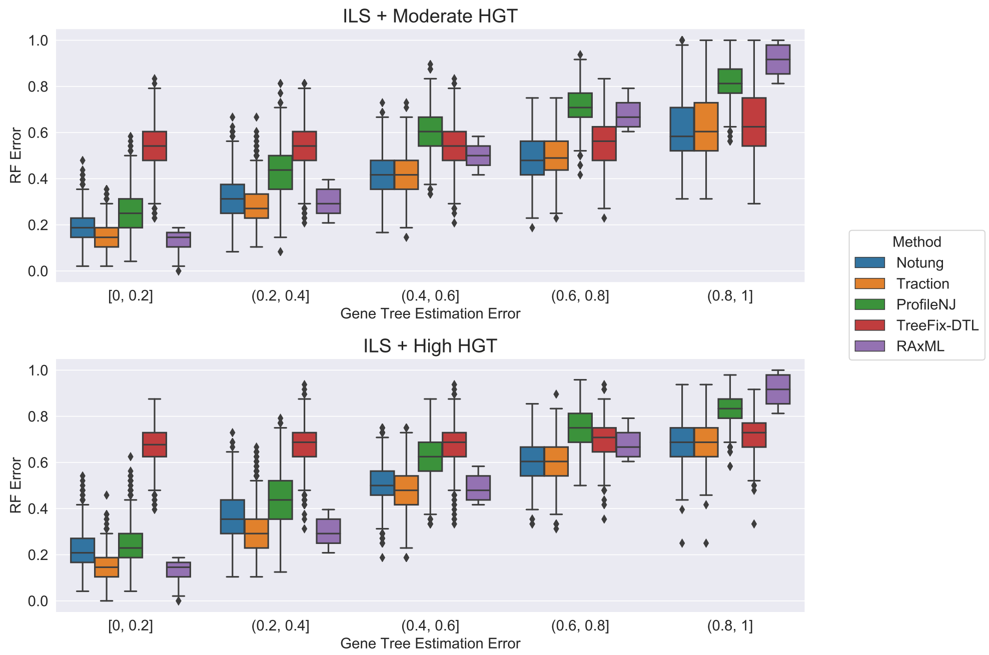


Figure 4.7: **Comparison of methods on ILS+HGT datasets with respect to average RF error rate as a function of GTEE.** Each boxplot displays the distribution of RF error across all replicates for a given method and level of GTEE. ecceTERA is not shown due to a high failure rate on this data.

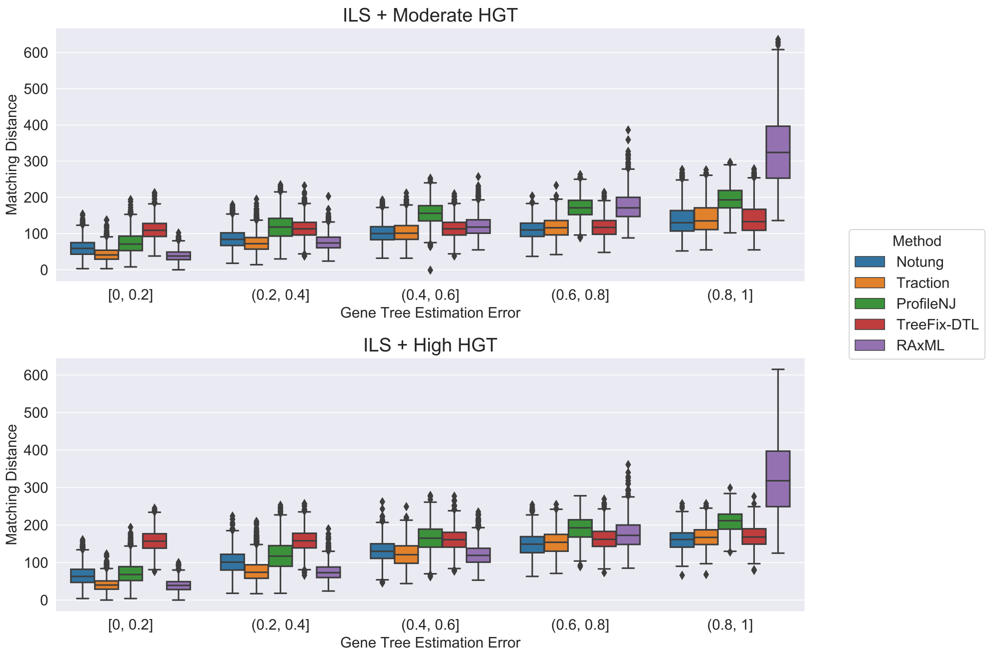


Figure 4.8: **Methods display similar relative accuracy when using matching distance to evaluate accuracy.** Boxplots show a comparison of methods on ILS+HGT datasets with respect to matching distance as a function of GTEE. ecceTERA is not shown due to a high failure rate on this data.

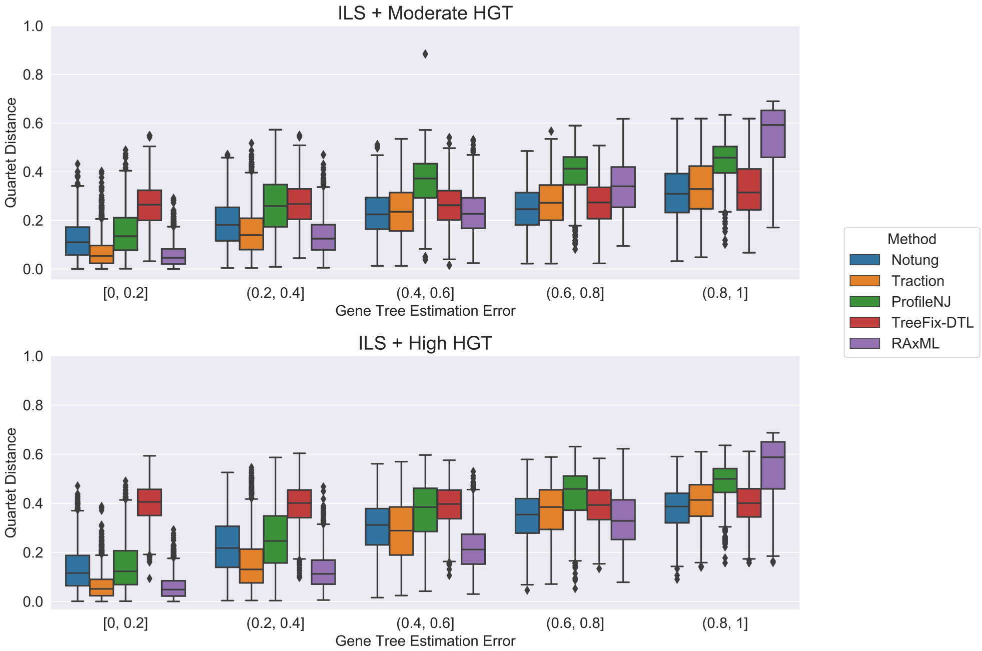


Figure 4.9: **Methods display similar relative accuracy when using quartet distance to evaluate accuracy.** Boxplots show a comparison of methods on ILS+HGT datasets with respect to quartet distance as a function of GTEE. ecceTERA is not shown due to a high failure rate on this data.

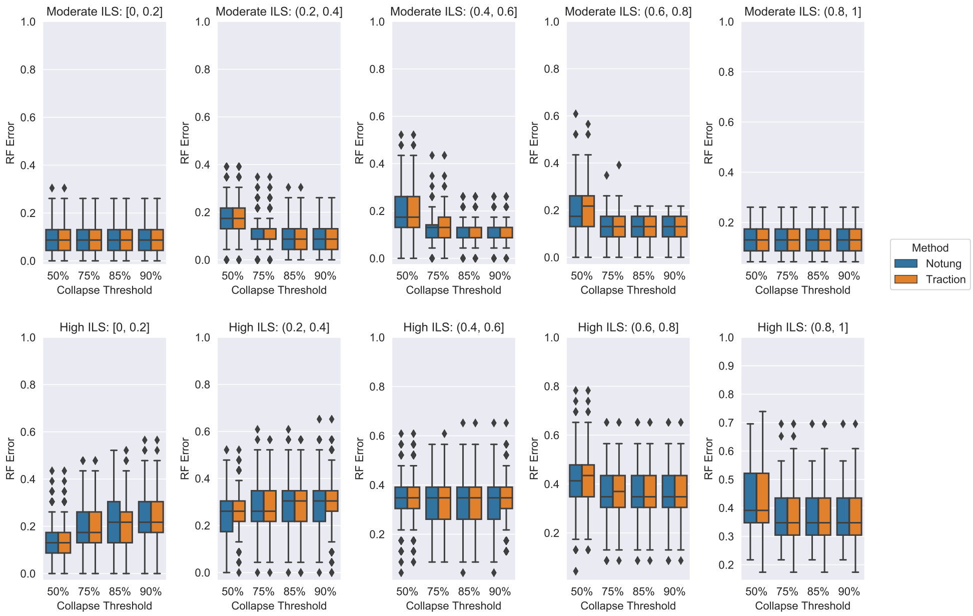


Figure 4.10: **TRACTION and Notung achieve similar performance across collapse thresholds.** Comparison of different collapse thresholds for gene trees on the ILS-only datasets. In each case, edges with support less than the threshold are collapsed before refinement. TRACTION and Notung completed in all instances, so no gene trees are removed.

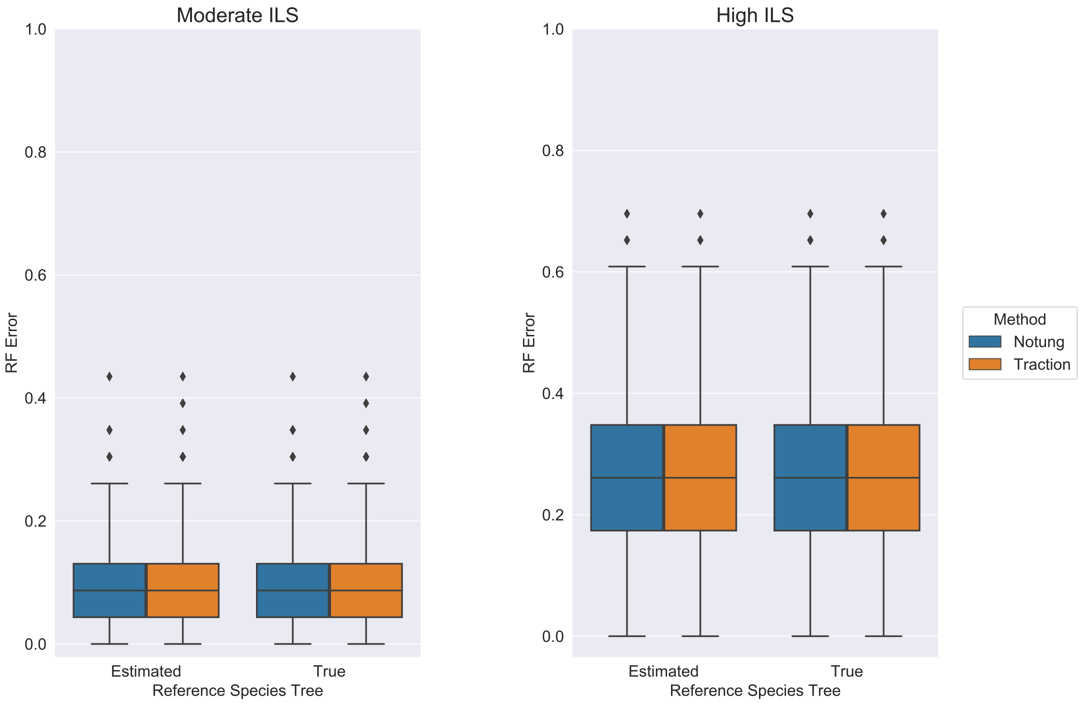


Figure 4.11: **TRACTION and Notung achieve similar performance when using a true species tree as reference.** Comparison of using a species tree estimated by ASTRID compared to the true species tree as a reference for gene trees on the ILS-only datasets. TRACTION and Notung completed in all instances, so no gene trees are removed.

Table 4.1: Empirical properties of the simulated datasets used in this study: GT-HET (gene tree heterogeneity, the average normalized RF distance between true gene trees and true species trees); GTEE (average gene tree estimation error); and the average distance of the ASTRID reference tree, to the true gene trees. The publications from which the simulated datasets are taken are also indicated. In total we analyzed 68,000 genes with varying levels and causes of true gene tree heterogeneity (to the true species tree) and GTEE. The ILS-only conditions each had 20 replicates, and the ILS+HGT conditions each had 50 replicates.

	GT-HET	GTEE	Distance ASTRID to true gene trees
ILS-only, Low ILS, 26 species [118]			
# sites varies	0.10	0.32	0.08
ILS-only, High ILS, 26 species [118]			
# sites varies	0.36	0.40	0.33
ILS+HGT, Moderate HGT (m5), 51 species [152]			
100 sites	0.54	0.63	0.55
250 sites	0.54	0.47	0.55
500 sites	0.54	0.47	0.54
ILS+HGT, High HGT (m6), 51 species [152]			
100 sites	0.68	0.62	0.68
250 sites	0.68	0.46	0.68
500 sites	0.68	0.38	0.68

Table 4.2: Total time (in seconds) for each method to correct 50 gene trees with 51 species on one replicate (label 01) of the HGT+ILS dataset with moderate HGT and sequences of length 100bp.

Method	Time (s)
EcceTERA	1470
NOTUNG	43
TRACTION	30
ProfileNJ	87
TreeFix-DTL	188

CHAPTER 5: PRIORITIZING ALTERNATIVE TUMOR PHYLOGENIES USING MUTATIONAL SIGNATURES WITH PHYSIGS

We now switch our focus from species to tumor phylogenies. We introduce a method, PhySigs, which infers changes in exposure to well-known mutational processes over the lifetime of a tumor. As a consequence, we improve upon the output of tumor phylogeny estimation methods by prioritizing phylogenies with more parsimonious exposures. Figures appear at the end of this chapter in Section 5.7.

5.1 INTRODUCTION

To understand the mechanisms by which mutations accumulate, researchers search large databases of somatic mutations and identify *mutational signatures*, patterns of mutations associated with distinct mutational processes across different types of cancer [13]. In addition to elucidating tumorigenesis, mutational signatures have found clinical applications [157]. One promising application is using a tumor's *exposure* to a signature associated with perturbed DNA damage repair as a biomarker for response to an established therapy, potentially increasing the number of patients who could benefit beyond standard driver-based approaches [158]. Methods for inferring the mutational signatures active in a given tumor are key to realizing this goal. However, initial analyses overlook *intra-tumor heterogeneity*, the presence of multiple clones with distinct complements of mutations that may be characterized by distinct mutational signatures. Here, we propose to study the dynamics of exposures to mutational signatures of clones within a tumor, in order to better understand tumorigenesis, accurately construct tumor phylogenies, and move towards devising more effective treatment plans.

A clone may be distinguished from its parental clone in a tumor phylogeny by a unique set of introduced mutations that appear in the clone but not in its parent. Introduced mutations provide a record of the mutational signatures acting on the clone at a particular location and time. Previous work to identify exposures to known mutational signatures can be classified in four broad categories. An initial body of work [12, 159, 160, 161] aimed to identify a single distribution of mutational signatures for all clones of a tumor, which we refer to as

This chapter contains material previously published in "PhySigs: Phylogenetic Inference of Mutational Signature Dynamics," which was presented at the 2020 *Pacific Symposium on Biocomputing* [156]. This work was done in conjunction with M. Leiserson and M. El-Kebir; MEK, SC conceived the project; SC, MEK designed and implemented the algorithm; ML implemented the visualization tool; SC, MEK established theory; SC, MEK performed experiments; SC wrote proof; SC, ML produced the figures; SC, MEK, ML contributed to the writing of this chapter.

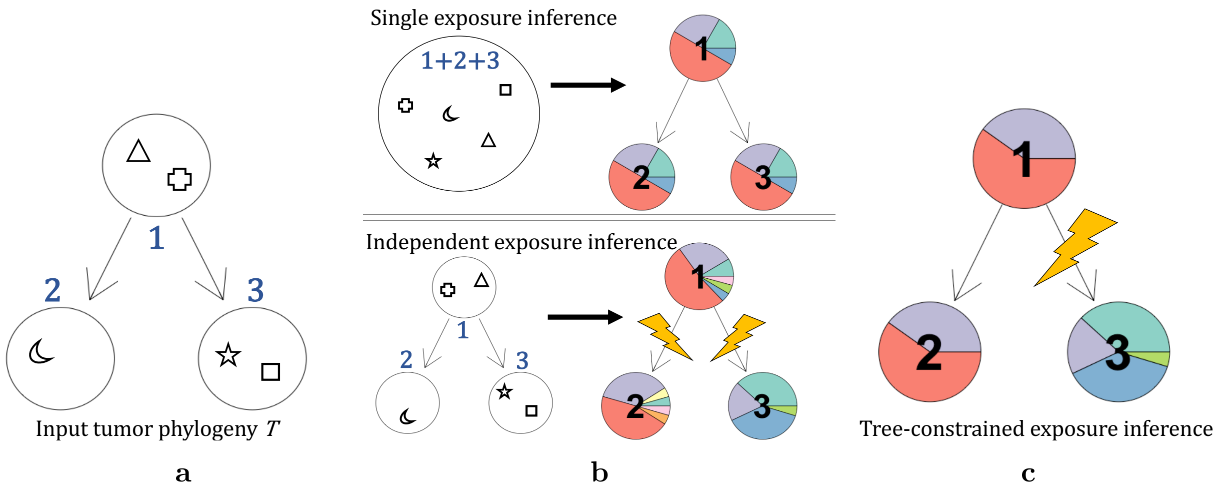

Input tumor phylogeny T
1
2
3
Single exposure inference
1+2+3
Independent exposure inference
Tree-constrained exposure inference
a
b
c

Figure 5.1: **PhySigs unites previous work on inference of clonal mutational signature exposures into one statistical framework by incorporating evolutionary context.** (a) The input is a tumor phylogeny T with nodes representing clones in a patient tumor, and the set of mutations (indicated by shapes) introduced in each clone. (b) Previous work generally falls into two categories and disregards evolutionary structure: While in single exposure inference all mutations are combined into one set for signature exposure inference, signatures for each clone are estimated independently in independent exposure inference. (c) Both previous problems are special cases of the problem solved by PhySigs, which incorporates evolutionary context to return a set of exposure shifts (lighting bolts) as well as the signature exposures for each cluster defined by the shifts.

single exposure inference (Fig. 5.1). This was followed by work [87, 162] that considered the distribution of exposures for each clone independently, called *independent exposure inference*. In addition to considering clones independently, Jamal-Hanjani et al. [87] clustered mutations into two categories: *clonal* mutations that are present in all clones vs. *subclonal* mutations that are present in only a subset of clones. Finally, Rubanova et al. [163] incorporated even more structure by studying the changes in exposures within a linear ordering of the clones. A similar idea to study the dynamics of APOBEC signature exposure has been explored experimentally in cell lines and patient-derived xenografts [164].

We build upon this line of work by proposing a model of clonal exposures that explicitly incorporates the tumor phylogeny relating clones. As new mutations interfere with key DNA repair pathways or carcinogenic environmental factors are added or removed, we would expect to see a change in the corresponding exposures along edges of the tumor phylogeny. Such *exposure shifts* induce a partition of the set of clones into disjoint clusters, where within each cluster the clones are ascribed the same set of relative signature exposures. To identify exposure shifts, we formulate the *Tree-constrained Exposure* (TE) problem (Fig. 5.1). We provide an algorithm, PhySigs, which solves the TE problem and provides a principled way

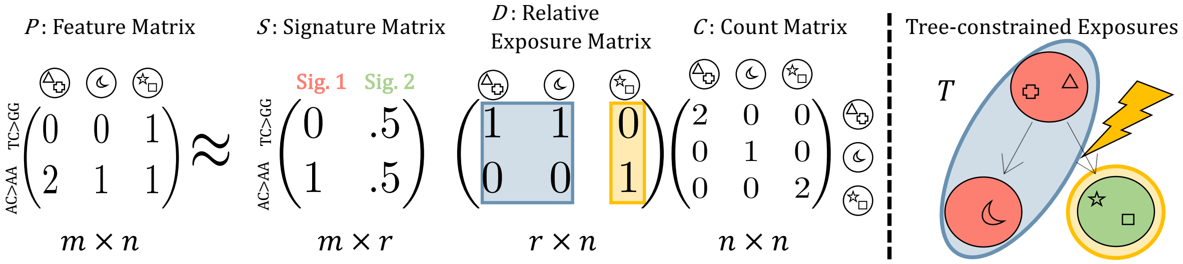


Figure 5.2: **PhySigs solves the TE problem for all combinations of exposure shifts.** PhySigs takes as input P, S, C and T and solves the TE problem, identifying for each value of $k \in \{1, \ldots, n\}$ a relative exposure matrix D composed of k identical columns corresponding to clusters of clones with identical exposures (denoted by blue and yellow). Edges between these clusters in T are interpreted to be where exposure shifts occurred (denoted by a lightning bolt). PhySigs uses the Bayesian Information Criterion to select the number k^* of clusters that best explain the data (here $k^* = 2$).

to select the number of exposure shifts such that the mutational patterns observed at each clone are accurately reconstructed without overfitting. PhySigs interpolates between single exposure inference and independent exposure inference, thus uniting previous work under one statistical framework. On simulated data, we demonstrate that PhySigs accurately recovers exposures and shifts. While PhySigs does not detect any exposure shifts in ovarian cancer [162], it identifies several exposure shifts in non-small-cell lung cancers [87], including at least one case with strong support from a driver mutation in the corresponding subclone. Moreover, PhySigs enables one to prioritize alternative, equally-plausible phylogenies inferred from the same input DNA sequencing data.

5.2 PROBLEM STATEMENT

We consider n samples with SNVs classified into the $m = 96$ mutation categories most commonly used for mutation signature analysis [13]. We assume each sample's SNVs are the product of r signatures of underlying mutational processes. The $m \times n$ *feature matrix* $P = [p_{ij}]$ indicates the number of mutations of category i in sample j. The $m \times r$ *signature matrix* $S = [s_{i\ell}]$ describes the probability signatures ℓ generate mutations of category i. The $r \times n$ *exposure matrix* $E = [e_{\ell j}]$ contains the number of mutations generated by signatures ℓ in samples j. The three matrices are related as follows:

$$P \approx SE. \tag{5.1}$$

Beginning with Alexandrov et al. [13], initial efforts to discover *de novo* mutational signatures shaping cancer genomes used non-negative matrix factorization. These efforts produced a compendium of 30 validated mutational signatures distributed by the Catalogue of Somatic Mutations in Cancer, [165] and researchers used the signature exposures to reveal signature etiology (e.g. Kim et al. [166]) and other applications (e.g. Trucco et al. [167] and Davies et al. [158]). However, these initial analyses disregard the clonal architecture of individual tumors. To understand the *clonal dynamics of mutational signatures*, we wish to identify signature exposures of the mutations that were introduced in each individual clone.

We start by recognizing that the exposures for each clone are proportional to the number of mutations present in the clone. We formalize this by defining a *relative exposure matrix* $D \in [0,1]^{r \times n}$, a matrix with nonnegative entries between 0 and 1. The relative exposure matrix D corresponding to an absolute exposure matrix E and feature matrix P is obtained by dividing the entries of each column j of E by the total number of mutations in the corresponding sample of P. In other words, we may view exposure matrix E as the product DC of a relative exposure matrix D and a diagonal *count matrix* $C \in \mathbb{N}^{n \times n}$ whose diagonal entries c_{jj} equal the number $\sum_{i=1}^{m} p_{ij}$ of mutations in clone j.

The first problem that we formulate assumes that the mutations introduced in every clone result from the same relative exposures.

Problem 5.1 (Single Exposure (SE))**.** Given feature matrix P, corresponding count matrix C and signature matrix S, find relative exposure matrix D such that $||P - SDC||_F$ is minimum and D is composed of identical columns.

Current methods [12, 159, 160, 161] implicitly solve this problem by estimating signatures of a single sample with mutations pooled across clones, as we will show in Section 5.3.

By contrast, the second problem assumes that the mutations introduced in each clone result from distinct exposures. In other words, we assume independence between the clones, leading the to the following problem.

Problem 5.2 (Independent Exposure (IE))**.** Given feature matrix P, corresponding count matrix C and signature matrix S, find relative exposure matrix D such that $||P - SDC||_F$ is minimum.

We note that the above problem is equivalent to the problem solved by current methods for patient-specific exposure inference [12, 159, 160, 161] where one replaces patients by clones, as was recently done by Jamal-Hanjani et al. [87]

An *exposure shift* is a significant shift in relative exposures of signatures between two clones. Recognizing that exposure shifts occur on a subset of the edges of a phylogeny T

(Fig. 5.1), we propose the following tree-constrained inference problem, which generalizes both previous problems (Fig. 5.2).

Problem 5.3 (Tree-constrained Exposure (TE))**.** Given feature matrix P, corresponding count matrix C, signature matrix S, phylogenetic tree T and integer $k \geq 1$, find relative exposure matrix D such that $||P - SDC||_F$ is minimum and D is composed of k sets of identical columns, each corresponding to a connected subtree of T.

We note that both SE and IE are special cases of TE, where $k = 1$ and $k = n$, respectively (Fig. 5.1). Moreover, the three problems are identical for a feature matrix P composed of a single clone ($n = 1$). Finally, for a fixed selection of k subtrees, the TE problem decomposes into k SE instances.

5.3 METHODS

5.3.1 Solving the SE problem

To solve the clone-specific exposure inference problems defined in the previous section, we wish to leverage current methods for patient-specific exposure inference. These current methods [12, 159, 160, 161] solve the problem of identifying absolute exposures $\mathbf{e}^* \in \mathbb{R}^r_{\geq 0}$ of a single patient minimizing $||\mathbf{q} - S\mathbf{e}||_F$ given feature vector $\mathbf{q}$ and signature matrix S, as described in Eq. (5.1).

In the following, we show how to reduce, in polynomial time, any SE instance (P, S) to the patient-specific instance $(\mathbf{q}, S)$. Specifically, we transform feature matrix $P = [p_{ij}]$ composed of $n \geq 1$ clones to a single-clone feature vector $\mathbf{q} = [q_i]$ by setting

$$q_i = \sum_{j=1}^{n} c_j \cdot p_{ij} \quad \forall i \in [m], \tag{5.2}$$

where c_j is the number $\sum_{i=1}^{m} p_{ij}$ of mutations introduced in clone j. Let N be the sum of the number of mutations in each sample squared, i.e.

$$N = \sum_{j=1}^{n} \left(\sum_{i=1}^{m} p_{ij} \right)^2 = \sum_{j=1}^{n} c_j^2. \tag{5.3}$$

We claim that relative exposure matrix D composed of n identical vectors $\mathbf{d}^*$ defined as $\mathbf{d}^* = \mathbf{e}^*/N$ is an optimal solution to SE instance (P, S).

Lemma 5.1. Let (P, S) be an instance of SE. Let $(\mathbf{q}, S)$ be the corresponding patient-specific instance with optimal solution $\mathbf{e}^*$. Then the relative exposure matrix D composed of n identical vectors $\mathbf{d}^* = \mathbf{e}^*/N$ is an optimal solution to SE instance (P, S).

Proof. We define $\mathbf{c} = [c_j]$ as an n-dimensional row vector, where c_j is the number of mutations introduced in clone j. We begin with (P, S), an arbitrary instance of SE, where we wish to find a vector $\mathbf{d}^*$ that equals

$$\underset{\mathbf{d}}{\operatorname{argmin}} ||P - S\mathbf{dc}||_F = \underset{\mathbf{d}}{\operatorname{argmin}} \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |p_{ij} - \sum_{\ell=1}^{r} s_{i\ell} \cdot d_\ell \cdot c_j|^2} \tag{5.4}$$

We now rearrange this equation to reflect the minimization problem for $(\mathbf{q}, S)$, the corresponding patient-specific exposure instance as described above. In doing so, we will show that the set of optimal solutions for $(\mathbf{q}, S)$ has the claimed relationship to the set of optimal solutions for (P, S). We start by squaring and then defining $\hat{p}_{ij} = \sum_{\ell=1}^{r} s_{i\ell} \cdot d_\ell \cdot c_j$ as the reconstructed value for feature i and sample j.

$$\underset{\mathbf{d}}{\operatorname{argmin}} \sum_{i=1}^{m} \sum_{j=1}^{n} |p_{ij} - \sum_{\ell=1}^{r} s_{i\ell} \cdot d_\ell \cdot c_j|^2 = \underset{\mathbf{d}}{\operatorname{argmin}} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(p_{ij}^2 - 2p_{ij}\hat{p}_{ij} + \hat{p}_{ij}^2\right) \tag{5.5}$$

We now distribute the inner sum.

$$\underset{\mathbf{d}}{\operatorname{argmin}} \sum_{i=1}^{m} \left[\sum_{j=1}^{n} p_{ij}^2 - 2(\sum_{j=1}^{n} p_{ij}\hat{p}_{ij}) + \sum_{j=1}^{n} \hat{p}_{ij}^2\right] \tag{5.6}$$

Next, we remove the first term, which is a constant, followed by substituting $\mathbf{d}$ with $\mathbf{e} = N \cdot \mathbf{d}$.

$$\underset{\mathbf{d}}{\operatorname{argmin}} \sum_{i=1}^{m} \left[-2(\sum_{j=1}^{n} p_{ij}\hat{p}_{ij}) + \sum_{j=1}^{n} \hat{p}_{ij}^2\right] = \frac{1}{N} \underset{\mathbf{e}}{\operatorname{argmin}} \sum_{i=1}^{m} \left[-2(\sum_{j=1}^{n} p_{ij}\hat{p}_{ij}) + \sum_{j=1}^{n} \hat{p}_{ij}^2\right] \tag{5.7}$$

We update $\hat{p}_{ij}$ terms using $\mathbf{e}$. Let $\hat{q}_i = \sum_{\ell=1}^{r} s_{i\ell} \cdot e_\ell$ be the reconstructed value for mutation category i where $e_\ell = d_\ell \cdot N$. Observe that $\hat{p}_{ij} = \sum_{\ell=1}^{r} s_{i\ell} \cdot d_\ell \cdot c_j = c_j \sum_{\ell=1}^{r} s_{i\ell} \cdot e_\ell / N = c_j \cdot \hat{q}_i / N$.

$$\frac{1}{N} \underset{\mathbf{e}}{\operatorname{argmin}} \sum_{i=1}^{m} \left[-2(\sum_{j=1}^{n} p_{ij} \frac{c_j \cdot \hat{q}_i}{N}) + \sum_{j=1}^{n} \left(\frac{c_j \cdot \hat{q}_i}{N}\right)^2\right] \tag{5.8}$$

We multiply inside the argmin by the positive constant $N > 0$, canceling terms using (5.3).

$$\begin{aligned} &\frac{1}{N} \operatorname*{argmin}_{\mathbf{e}} \sum_{i=1}^{m} \left[-2(\sum_{j=1}^{n} p_{ij} \cdot c_j \cdot \hat{q}_i) + \hat{q}_i{}^2 \frac{\sum_{j=1}^{n} c_j^2}{N} \right] \qquad (5.9) \\ &= \frac{1}{N} \operatorname*{argmin}_{\mathbf{e}} \sum_{i=1}^{m} \left[-2(\sum_{j=1}^{n} p_{ij} \cdot c_j \cdot \hat{q}_i) + \hat{q}_i{}^2 \right] \qquad (5.10) \end{aligned}$$

We add back in a constant term.

$$\frac{1}{N} \operatorname*{argmin}_{\mathbf{e}} \sum_{i=1}^{m} \left[(\sum_{j=1}^{n} c_j \cdot p_{ij})^2 - 2(\sum_{j=1}^{n} c_j \cdot p_{ij})\hat{q}_i + \hat{q}_i^2 \right] \tag{5.11}$$

We now substitute in for $\mathbf{q}$ following (5.2).

$$\frac{1}{N} \operatorname*{argmin}_{\mathbf{e}} \sum_{i=1}^{m} \left(q_i^2 - 2q_i\hat{q}_i + \hat{q}_i^2 \right) = \frac{1}{N} \operatorname*{argmin}_{\mathbf{e}} \sum_{i=1}^{m} |q_i - \hat{q}_i|^2 \tag{5.12}$$

Finally, we substitute out $\hat{\mathbf{q}}$.

$$\begin{aligned} \frac{1}{N} \operatorname*{argmin}_{\mathbf{e}} \sum_{i=1}^{m} \left| q_i - \sum_{\ell=1}^{r} s_{i\ell} \cdot e_\ell \right|^2 &= \frac{1}{N} \operatorname*{argmin}_{\mathbf{e}} \sqrt{\sum_{i=1}^{m} \left| q_i - \sum_{\ell=1}^{r} s_{i\ell} \cdot e_\ell \right|^2} \qquad (5.13) \\ &= \frac{1}{N} \operatorname*{argmin}_{\mathbf{e}} ||\mathbf{q} - S\mathbf{e}||_F = \frac{\mathbf{e}^*}{N}. \qquad (5.14) \end{aligned}$$

Observe that this final equation contains the minimization problem for the patient-specific instance $(\mathbf{q}, S)$ as defined in our reduction. Thus, we get that $\mathbf{d}^* = \mathbf{e}^*/N$ as claimed. QED.

5.3.2 Solving the IE and TE problems

IE problem. In the IE problem, we are given a feature matrix P, a signature matrix S and seek a relative exposure matrix D such that $||P - SDC||_F$ is minimum. We solve this problem by decomposing it into n SE problem instances, each composed of a single clone. For each resulting SE instance, we use the reduction described in Section 5.3.1 to the patient-specific exposure problem.

TE problem. In the TE problem, we are given a feature matrix P, a signature matrix S, a phylogenetic tree T and an integer $k \geq 1$. The tree T has n nodes and thus $n - 1$

edges. To solve this problem, we exhaustively enumerate all $\binom{n-1}{k-1}$ combinations of $k-1$ edge removals that lead to k connected subtrees. Each connected subtree correspond to a single SE instance, which may be solved using the reduction to the patient-specific exposure problem described previously. We select the combination of $k-1$ edges that minimizes the objective function.

5.3.3 PhySigs

Model selection for k. To decide on the number k of subtrees to consider, we use the Bayesian Information Criterion (BIC)[168]. That is, we evaluate each optimal solution for each number $k \in \{1, \ldots, n\}$ of subtrees. For a fixed k, the number of observations equals the size of matrix P, i.e. mn, and the number of parameters equals the number of entries in unique columns of D, i.e. kr. Let $L(k) = \min_D ||P - SDC||_F$ be the optimal value for k subtrees, then the corresponding BIC value is

$$\mathrm{BIC}(L(k)) = mn \log(L(k)/(mn)) + kr \log(mn). \tag{5.15}$$

We select the number k that has the smallest BIC value.

PhySigs implementation. We implemented the above algorithm for the TE problem that includes model selection in R. Our method, PhySigs, uses deconstructSigs [159] as a subroutine for solving the underlying SE problems. We note that while deconstructSigs is a heuristic and does not solve the SE problem optimally, it was found by Huang et al. [160] to give comparable results to the optimal solution in most patients. The original PhySigs code is available at https://github.com/elkebir-group/PhySigs, and an R package implementation is available at https://github.com/elkebir-group/PhySigs_R.

5.4 EVALUATION AND RESULTS

Results were obtained on a laptop with a 2.9 GHz CPU and 16 GB RAM. All instances completed in minutes, with the exception of a 15 clone lung cancer instance taking hours.

5.4.1 PhySigs accurately recovers exposures and shifts in simulated data

We first assess PhySigs's ability to correctly identify model parameters for data generated under the Tree-constrained Exposure model. To do so, we generate simulated data with clone-specific mutations in $m = 96$ categories resulting from exposure to $r = 30$ COSMIC v2 signatures [165], comprising matrix S. Specifically, we simulate 20 phylogenetic trees T with $n \in \{5, 7\}$ clones, each clone containing between 20 and 200 mutations, as described by the count matrix C. For each phylogenetic tree T, we generate a partition of $k \in \{1, 2, 3\}$ connected subtrees, assigning each subtree a relative exposure vector $\mathbf{d}$ by drawing from a symmetric Dirichlet distribution (with concentration parameter $\alpha = 0.2$). For each combination of T and k, this yields a relative exposure matrix D. Next, we introduce Gaussian noise with mean $\mu = 0$ and standard deviation $\sigma \in \{0.1, 0.2, 0.3\}$, amounting to a $m \times n$ matrix X. Finally, we generate the feature matrix P as $SDC + X$. Thus, we have a total of 180 TE problem instances (T, P).

Fig. 5.3 shows that PhySigs identifies relative exposures D^* that are close to their corresponding simulated exposures D in varying noise regimes and simulated number of exposure shifts (Fig. 5.3a). Moreover, the number of exposure shifts is correctly identified (Fig. 5.3b), as well as their exact locations (Fig. 5.3c). In summary, our simulations demonstrate that PhySigs is robust to noise and is able to accurately reconstruct relative exposures and exposure shifts within this model.

5.4.2 PhySigs suggests the absence of exposure shifts in ovarian cancer

We run PhySigs on a ovarian cancer dataset [162] composed of 7 tumors, containing between 3 to 9 clones, each with a median of 468 mutations. We apply deconstructSigs's [159] trinucleotide normalization to correct feature matrix P by the number of times each trinucleotide is observed in the genome, as this is whole-genome sequencing data. We focus our attention on COSMIC v2 Signatures 1, 3, 5, which have been designated as occurring in ovarian cancer [169].

We find that PhySigs does not identify any exposure shifts (i.e. $k^* = 1$), assigning identical relative exposures to all clones within each patient (data not shown). This finding is corroborated when comparing PhySigs's inference error to the error obtained when solving the Independent Exposure (IE) problem, showing only a marginal decrease (median error of 149 for IE compared to 150 for PhySigs).

We see similar patterns in exposure shifts and content when additionally including *BRCA* associated signatures 2 and 13, as well as including all breast-cancer associated mutational signatures (1, 2, 3, 5, 6, 8, 10, 13, 17, 18, 20, 26 and 30).[b] It is known that ovarian cancer is predominantly driven by structural variants and copy number aberrations, which has recently motivated the use of copy number signatures rather than SNV signatures to study mutational patterns in ovarian carcinomas[170]. Indeed, examining the exposures, we find that these are dominated by Signatures 1 and 3—Signature 1 is a clock-like signature [171] and Signature 3 is highly correlated with clock-like Signature 5 (cosine similarity of 0.83). Thus, in the absence of evidence otherwise, PhySigs will not identify exposure shifts.

5.4.3 PhySigs identifies exposures shifts in a lung cancer cohort

Jamal-Hanjani et al. [87] reconstructed phylogenetic trees for 91 lung cancer patients, with 2 to 15 clones per patient (Fig. 5.4b). Here, we use PhySigs to study the clonal dynamics of mutational signatures in this cohort. Since these data have been obtained using whole exome sequencing, we use deconstructSigs' [159] exome normalization feature to correct feature matrix P. We restrict our attention to COSMIC v2 Signatures 1, 2, 4, 5, 6, 13 and 17, which are associated with non-small-cell lung carcinoma.[b]

PhySigs identifies exposure shifts in 20 out of 91 patients, with a single exposure shift in 16 patients and two exposure shifts in 4 patients (Fig. 5.4b). To understand why PhySigs identified exposures shifts in these 20 patients, we compare the error $||P - SDC||_F$ of PhySigs's solution to the Tree-constrained Exposure (TE) against the errors of solutions to the Single Exposure (SE) problem and the Independent Exposure (IE) problems. We find that the median error of the SE problem is 51, compared to 48 for TE/PhySigs and 47 for the IE problem (Fig. 5.4c). The decrease between SE and TE/IE suggests that enforcing a single exposure results is a poor explanation. On the other hand, the marginal decrease between TE and IE (48 vs. 47) suggests that the exposures inferred for each cloned independently by IE likely suffer from overfitting. Indeed, Fig. 5.4d shows the input to the IE problem instance is composed of clones with a median number of only 10 mutations, resulting in poorly supported clone-specific exposures.

We next sought to validate the exposures inferred by PhySigs by identifying branches of phylogenetic trees with a significant change in exposure that could be explained by other observations of the tumor. We reasoned that tumors with a subclonal mutation to a gene in the DNA mismatch repair (MMR) pathway could lead to a large increase in Signature 6 (previously associated with DNA mismatch repair [13]) along one branch of the tree. Indeed, we find one such example in the lung cancer dataset, which we illustrate in Fig. 5.5. According to the driver mutation classifications provided by Jamal-Hanjani et al. [87], PhySigs finds that one subclonal branch of the tree for CRUK0064 has a putative driver mutation to MMR gene *MLH1* and a high percentage of mutations from MMR-associated Signature 6 (60.7% of the 401 mutations). The remaining cancer cells outside this branch have zero Signature 6 exposure, supporting the claim that the mutation in *MLH1* is indeed driving the increase in Signature 6 exposure. We note that using an approach that does not fully incorporate a phylogenetic tree, such as the linearly ordering mutations by cancer cell fractions (CCFs) proposed by Rubanova et al.[163], may overlook exposure shifts that are only in one branch of the tree as signal may be drowned out by mutations in other branches with similar CCFs.

Jamal-Hanjani et al. [87] identified multiple trees for 25 out of 91 patients. This is due to the underdetermined nature of the phylogeny inference problem from bulk DNA sequencing samples [96, 109]. We show that PhySigs provides an additional criterion for prioritizing alternative phylogenetic trees. Patient CRUK0025 has two alternative trees, T_1 and T_2, each composed of $n = 7$ clones, with uncertainty in the placement of clone 5 (Fig. 5.6). Examining the error for varying number $k \in \{1, \ldots, n\}$ of subtrees (Fig. 5.6a), we find that T_1 (Fig. 5.6b) has smaller error than T_2 (Fig. 5.6c) for the selected number $k^* = 3$ of subtrees according to the BIC (1,106 for T_1 vs. 1,122 for T_2), with only two exposure shifts. Moreover, to achieve a similar error in tree T_2, three exposure shifts are required (Fig. 5.6d). Assuming the more parsimonious explanation is more likely for a fixed magnitude of error, PhySigs's optimization criterion enables the prioritization of alternative trees in the solution space, preferring T_1 over T_2 for this patient.

5.5 VISUALIZATION TOOL

We also implemented a tool to allow users to visualize exposure shifts and explore mutations in driver genes introduced along the edges of the tumor phylogeny. The integration of exposures with drivers facilitates the identification of subclonal drivers that precede related exposure shifts, such as the example depicted in Fig. 5.5. The PhySigs R package can produce the necessary exposure input, but the tool stands alone so that it may also be used with exposures inferred by existing or future methods. A picture of the interface is shown in Fig. 5.7. The tool is publicly available at https://physigs-tree-browser.herokuapp.com.

5.5.1 Input data

The input data to the tool consists of three files, which can be pasted into the web interface for display. The first file describes the the tumor phylogenies. Each row corresponds to a patient tree and includes the patient ID, the tree ID, the number of nodes in the tree, the nodes labels for the tree, and the edges comprising the tree. The second file describes the exposures to display at the nodes of each patient tree. Each row corresponds to a node in a patient tree and includes the patient ID, the tree ID, the node cluster ID, the node label, and the percent exposure to each signature of interest. Our R package contains convenient functions formatting the above input to the tool's user interface, but this information could be taken from other exposure inference tools as well.

The third file contains optional information about driver mutations present in the patient data. Each row corresponds to one of these mutations and includes the patient ID, the ID of the node where the mutation was first introduced, the name of the gene where the mutation occurred, the type of mutation (e.g., missense, nonsense), HGVSp annotation on protein sequence [172], mutation effect if known (e.g., Loss-of-function), and whether or not it is known to be oncogenic. These mutation descriptions are all optional, but are conveniently displayed for the user when provided. This will typically need to be collected from another data source such as VEP [173], COSMIC [165], and OncoKB [174]. In particular, we used the maf2maf software available at https://github.com/mskcc/vcf2maf#maf2maf, which uses VEP as a backend, and the OnkoKB annotator, available at https://github.com/oncokb/oncokb-annotator. See Table 5.1 for complete list of input fields and example inputs.

5.5.2 Tool Features

The visualization tool lets users view patient tumor phylogenies with mutational signature exposures overlaid on each node. In particular, each node of the tree is represented by a pie chart showing the percent inferred exposures at that node. When exposure shifts are provided by the user, the placement of exposure shifts and resulting node clusters are displayed. The top of the window contains a table giving the precise exposures to mutational signatures for each cluster of clones. Up to two alternative trees for a patient can be displayed side-by-side to facilitate comparisons.

When information about driver mutations is provided, the user can also see the number of mutations occurring in a driver gene on each edge. The user can then flag driver genes of interest, and these gene names will appear as a label on the edge corresponding to where that mutation was introduced. Finally, clicking on an edge opens a pop-up box that provides additional information about the driver mutations introduced on that edge, including the gene name, mutation type, HGVSp annotation [172], effect, and oncogenic status.

5.6 DISCUSSION

Based on the idea that exposures may change along edges of a tumor phylogeny, we introduced a model that partitions the tree into disjoint sets of clusters, where the clones within each cluster are ascribed the same set of relative signature exposures. Using this model, we formulated the *Tree-constrained Exposure* (TE) problem and provided an algorithm PhySigs that includes a principled way to select the number of exposure shifts such that the mutational patterns observed at each clone are accurately reconstructed without overfitting. PhySigs unites previous work under one statistical framework, interpolating between single exposure inference [12, 159, 160, 161] and independent exposure inference [87, 162]. Our simulations demonstrated that PhySigs accurately recovers exposures and shifts. On real data, we found that while PhySigs does not detect any exposure shifts in ovarian cancer [162], it identified several exposure shifts in non-small-cell lung cancers [87], at least one of which is strongly supported by an observed subclonal driver mutation in the mismatch repair pathway. In addition, we showed that PhySigs enables the prioritization of alternative, equally-plausible phylogenies inferred from the same input data.

There are several avenues of future work. First, the hardness of the TE problem remains open for the case where $k = O(n)$. Second, PhySigs exhaustively enumerates all 2^n partitions of the n nodes of input tree T. It will be worthwhile to develop efficient heuristics that return solutions with small error. Third, we plan to assess statistical significance of solutions returned by PhySigs using permutation tests or bootstrapping, similarly to Huang et al.[160] Fourth, building on our results of tree prioritization using PhySigs, we may use our model to resolve additional tree ambiguities such as polytomies (nodes with more than two children), akin to previous work in migration analysis of metastatic cancers[99]. Fifth, we plan to use our model to study population-level trajectories of clonal exposures to mutational signatures. Finally, population-level analysis of clone-specific mutations may lead to better identification of mutational signatures rather than the current tumor-level analysis [13].

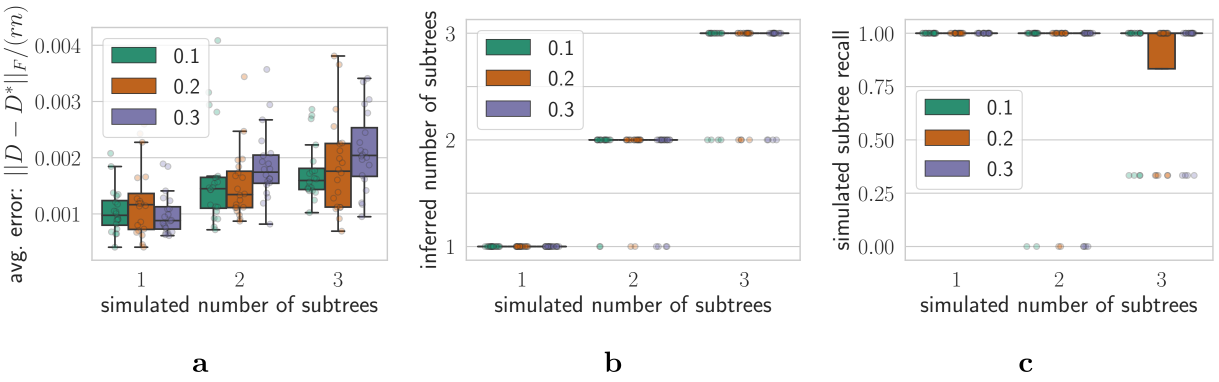


Figure 5.3: **Simulations show that PhySigs is robust to noise, accurately reconstructing simulated relative exposures as well as the number and location of exposure shifts.** Each boxplot contains results from 20 trees, colors indicate Gaussian noise standard deviations σ. (a) Error between the inferred and simulated relative exposure matrices. (b) The number of inferred vs. simulated subtrees. (c) The fraction of correctly recalled simulated subtrees.

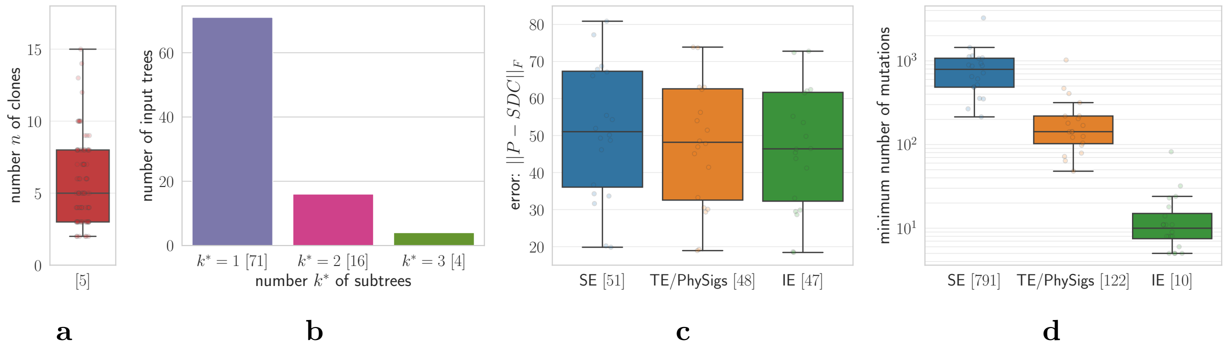


Figure 5.4: **PhySigs infers accurate exposures without overfitting in a lung cancer cohort of phylogenetic trees [87].** Median values of each box plot are in square brackets. (a) This cohort contains 91 patients with 2 to 15 clones. (b) PhySigs partitions the trees into $k^* \in \{1, \ldots, 3\}$ subtrees, solving the Tree-constrained Exposure (TE) problem and selecting k^* following the Bayesian Information Criterion. (c) The relative exposure matrix D inferred by PhySigs has smaller error compared to solving the Single Exposure (SE) problem, and comparable error to solving the Independent Exposure (IE) problem. (d) The latter results in overfitting as evidenced by the small number of mutations in the smallest cluster (median of 10 [green] vs. 122 for PhySigs [orange]).

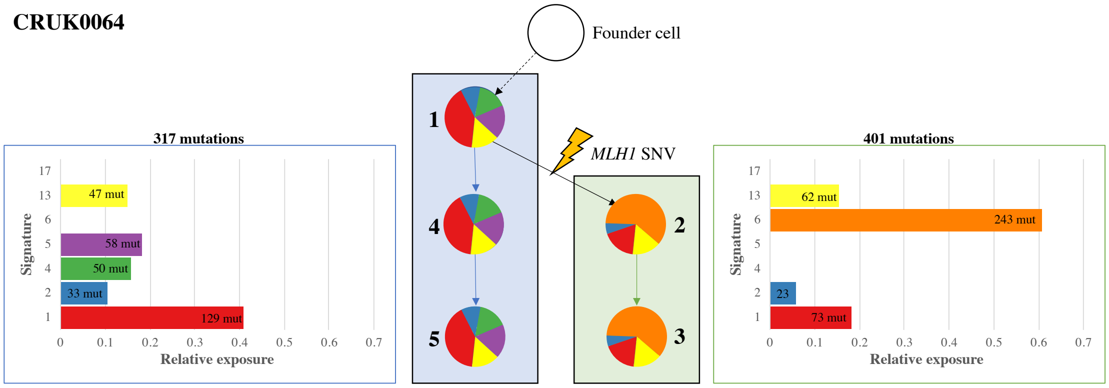
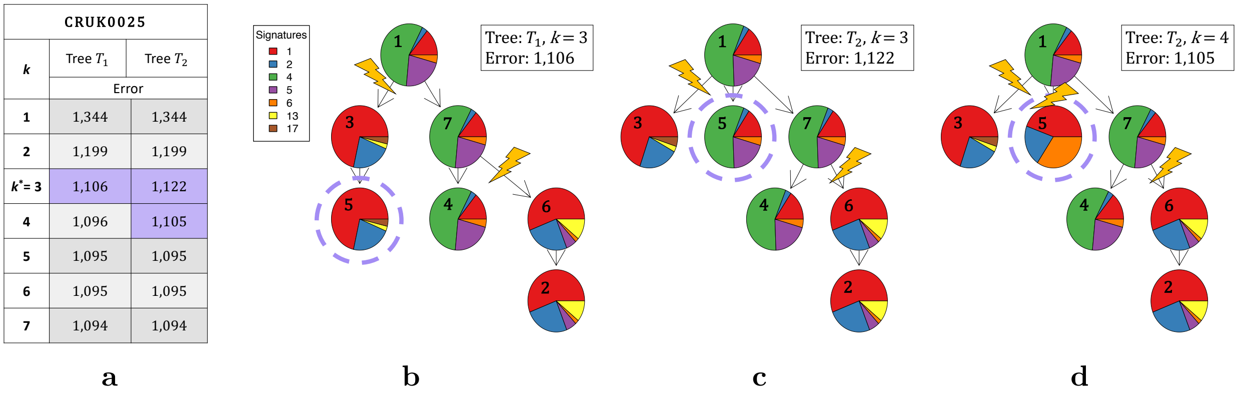


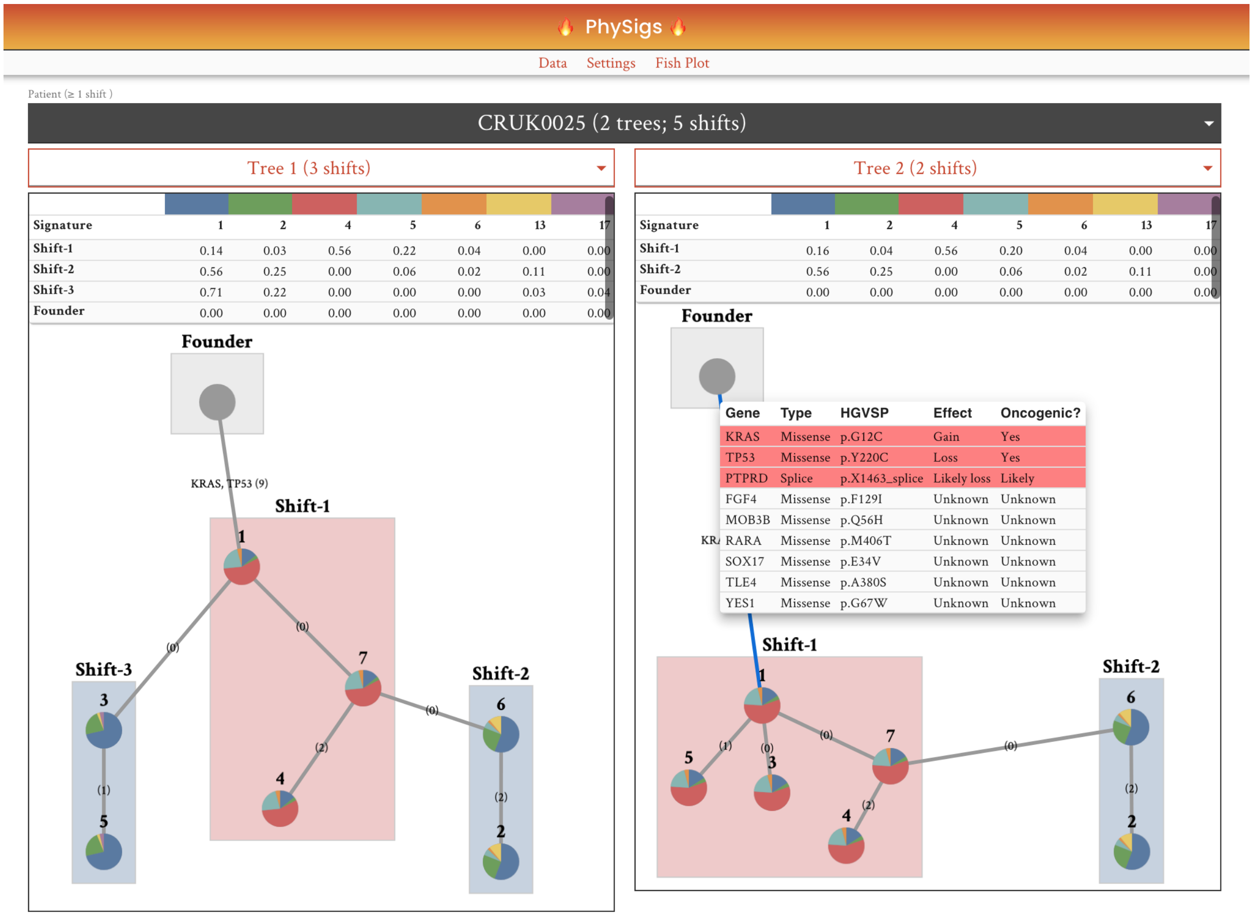
Figure 5.5: **PhySigs detects a large increase in DNA mismatch repair-associated Signature 6 (orange) along one branch (clusters 2 and 3; green) of the CRUK 0064 tree.** In support of this finding, the branch includes a subclonal driver mutation to DNA mismatch repair gene *MLH1*.



CRUK0025

k	Tree T_1	Tree T_2
	Error	
1	1,344	1,344
2	1,199	1,199
$k^* = 3$	1,106	1,122
4	1,096	1,105
5	1,095	1,095
6	1,095	1,095
7	1,094	1,094

Figure 5.6: **PhySigs provides an additional criterion for prioritizing tumor phylogenies when multiple solutions exist.** Jamal-Hanjani et al. [87] identified two potential tumor phylogenies for lung cancer patient CRUK0025 with discrepancies in the placement of clone 5. (a) For each tree, we show the minimum error identified by PhySigs for all k. (b) The optimal exposures inferred by PhySigs for tree T_1 for $k = 3$. Note that this solution was selected by the BIC. (c) The optimal exposures inferred by PhySigs for tree T_2 for $k = 3$. With the same number of exposure shifts, T_2 results in a higher error than T_1. (d) Three exposure shifts ($k = 4$) in T_2 are necessary to achieve the same level of error as two exposure shifts in T_1, suggesting that T_1 is the more accurate tree reconstruction.



Signature	1	2	4	5	6	13	17
Shift-1	0.14	0.03	0.56	0.22	0.04	0.00	0.00
Shift-2	0.56	0.25	0.00	0.06	0.02	0.11	0.00
Shift-3	0.71	0.22	0.00	0.00	0.00	0.03	0.04
Founder	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Signature	1	2	4	5	6	13	17
Shift-1	0.16	0.04	0.56	0.20	0.04	0.00	0.00
Shift-2	0.56	0.25	0.00	0.06	0.02	0.11	0.00
Founder	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Gene	Type	HGVSP	Effect	Oncogenic?
KRAS	Missense	p.G12C	Gain	Yes
TP53	Missense	p.Y220C	Loss	Yes
PTPRD	Splice	p.X1463_splice	Likely loss	Likely
FGF4	Missense	p.F129I	Unknown	Unknown
MOB3B	Missense	p.Q56H	Unknown	Unknown
RARA	Missense	p.M406T	Unknown	Unknown
SOX17	Missense	p.E34V	Unknown	Unknown
TLE4	Missense	p.A380S	Unknown	Unknown
YES1	Missense	p.G67W	Unknown	Unknown

Figure 5.7: **Publicly hosted visualization tool showing example patient CRUK0025.** This tool allows users to compare alternative phylogenies for a given patient (Tree 1 on left and Tree 2 on right). When there are more than two trees, a drop-down menu lets the user select two trees to compare at a time. For each tree, the user sees the exposures inferred for each clone along with the clusters of clones induced by exposure shifts. Each edge of the tree is labeled by the number of mutations occurring in driver genes along with user-specified genes of interest (e.g., KRAS, TP53). Double clicking on an edge reveals more information about these mutations, including the gene name, mutation type, HGVSp annotation, effect, and oncogenic status. Finally, the trees are interactive and can be dragged by the user to rearrange the nodes for more convenient viewing.

Table 5.1: A description of the input fields for the exposure visualization tool. While this can be used with the exposures output by PhySigs, it is possible to provide the tool with exposures inferred by any method.

Variable	Description	Example
Patient Tree Data Fields (Required)		
patient	Unique ID for patient	CRUK0006
tree	Unique ID for patient tree within patient tree set	1
num_nodes	Number of nodes in tree	2
nodes	Node labels in tree delimited by semicolon	1; 2; 3
edges	Edges in tree indicated by parent arrow child and delimited by semicolon	1->2;1->3
Clonal Exposure Data Fields (Required)		
patient	Unique ID for patient	CRUK0006
tree	Unique ID for patient tree within patient tree set	1
k	Label for exposure cluster node is assigned to in tree. Set default to 0 if not relevant.	0
node	Node label in tree	3
Signature.i	Percent exposure at node to mutational signature i. Include column for each relevant signature.	0.23
Driver Mutation Data Fields (Not Required)		
patient	Unique ID for patient	CRUK0006
node	Node label in tree	3
gene	Gene name where mutation occurred. Any classification system is accepted.	TP53
variant_class	Type of mutation. Any type is accepted.	Splice_Site
hgvsp	HGVSp protein annotation	p.R280K
mutation_effect	Mutation effect on protein. Set default to Unknown.	Loss-of-function
oncogenic	"The ability to induce or cause cancer" as defined in The Biology of Cancer by Robert Weinberg (2014). Set default to Unknown.	Oncogenic

CHAPTER 6: PRIORITIZING ALTERNATIVE TUMOR PHYLOGENIES USING CANCER PATIENT COHORTS WITH RECAP

In the previous chapter, we looked at how mutational signatures could be leveraged to prioritize alternative, equally-plausible tumor phylogenies inferred for the same patient. Here, we propose a method, RECAP, which uses data from cohorts of cancer patients, rather than mutational signatures, to similarly reduce the solution space by selecting the tree most consistent with other patients. In doing so, we also learn about the patterns of evolution shared across different clusters of patient tumors. Proofs and figures appear at the end of this chapter in Sections 6.6 and 6.7, respectively.

6.1 INTRODUCTION

The grouping of cancer patients into subtypes with similar patterns of evolution holds the potential to enhance current pathology-based subtypes, thereby improving our understanding of tumorigenesis and leading to better stratification of tumors with respect to survival and response to therapy [88, 89, 90, 91]. However, the two types of current sequencing technologies, bulk and single-cell DNA sequencing, each present unique challenges to the task of identifying repeated evolutionary trajectories. With bulk DNA sequencing, the input is a mixed sample composed of sequences from potentially millions of different cells that must be deconvolved [95, 96]. With single-cell DNA sequencing, the input has elevated rates of false positives, false negatives, and missing data [110]. Specialized tumor phylogeny inference methods must be used to analyze these data, but current methods infer many plausible trees for the same input, leading to large solution spaces of phylogenies with different mutation orderings. Importantly, alternative phylogenies at the individual patient level obfuscate repeated patterns of cancer evolution at the cohort level.

Two recent methods, REVOLVER [90] and HINTRA [91], propose to select one phylogeny for each patient so that the resulting trees are maximally similar, enabling the identification of repeated evolutionary trajectories. There are several limitations to these existing approaches. First, since HINTRA [91] exhaustively enumerates all possible (directed) two-state perfect phylogenies, which grows as n^{n-1} where n is the number of mutations, it does

This chapter contains material previously presented at the 2020 *European Conference on Computational Biology* [175] and later published in *Bioinformatics* under the title "Detecting Evolutionary Patterns of Cancers using Consensus Trees." This work was done in conjunction with J. Kim, N. Chia, O. Koyejo, and M. El-Kebir; MEK conceived the project; SC, MEK designed and implemented the algorithm; SC, MEK, JK, OK, NC established theory; SC, MEK proved computational hardness; SC proved dynamic programming correctness; SC, JK, MEK performed experiments; SC, MEK produced the figures; SC, MEK contributed to the writing of this chapter.

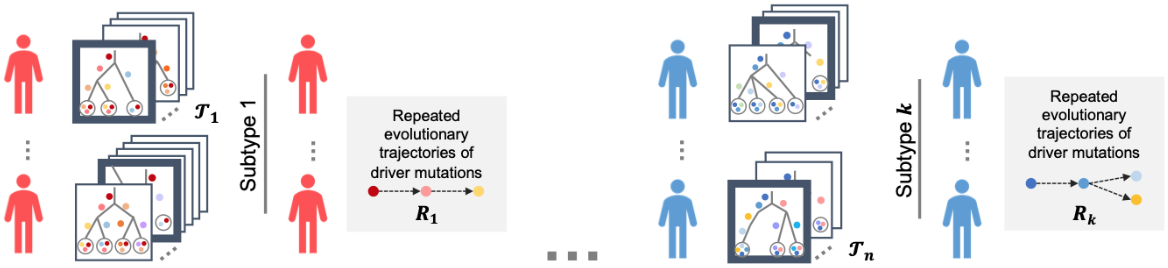

$\mathcal{T}_1$
Subtype 1
Repeated evolutionary trajectories of driver mutations
R_1
Subtype k
Repeated evolutionary trajectories of driver mutations
R_k
$\mathcal{T}_n$

Figure 6.1: **RECAP solves the Multiple Choice Consensus Tree problem.** Given a family $\{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$ of sets of patient trees, we simultaneously cluster n patients into k subtypes of evolutionary trajectories $\{R_1, \ldots, R_k\}$ and select a phylogeny for each patient.

not scale beyond a small number $n = 5$ of mutations. Second, neither HINTRA [91] nor REVOLVER [90] directly account for the presence of distinct subtypes of patients with distinct evolutionary patterns. Specifically, neither method uses a mixture model to represent the selected patient trees, assuming all selected trees to originate from a single distribution. REVOLVER tries to recover a patient clustering only after the fact, i.e. hierarchical clustering is performed only after inference of the selected trees and their single generating distribution. This is a serious limitation of both methods as the presence of distinct subtypes with distinct evolutionary trajectories is a documented phenomenon in cancer [88, 89, 176].

Here, we view the problem of identifying repeated patterns of tumor evolution as a consensus tree problem, where the consensus tree summarizes different patient phylogenies. Leveraging previous work on the Multiple Consensus Tree (MCT) problem [177], we formulate the Multiple Choice Consensus Tree (MCCT) optimization problem to simultaneously (i) select a phylogeny for each patient in a cancer cohort, (ii) cluster the patients to account for subtype heterogeneity, and (iii) identify a representative consensus tree for each patient cluster (Fig. 6.1). We prove the problem to be NP-hard. We then introduce Revealing Evolutionary Consensus Across Patients (RECAP), a coordinate ascent algorithm as a heuristic for solving this problem. We also include a model selection criterion for identifying the number k of subtypes needed to explain a dataset. On simulated data, we show that RECAP outperforms existing methods that do not support diverse evolutionary trajectories. We demonstrate the use of RECAP on real data, identifying well-supported evolutionary trajectories in a non-small cell lung cancer cohort and a breast cancer cohort.

6.2 PROBLEM STATEMENT

Recall that we represent the evolutionary history of a tumor by a rooted tree T whose root vertex is denoted by $r(T)$, vertex set by $V(T)$, and directed edge set by $E(T)$. We represent each non-root vertex $v \neq r(T)$ by the mutations $\mu(v) = \mu(u, v)$ introduced on its unique incoming edge (u, v). Consistent with most current phylogenetic analyses in cancer genomics, this work adheres to the ISA, i.e. each mutation is gained exactly once and is never subsequently lost. Throughout this chapter, we will refer to rooted trees adhering to the ISA simply as trees.

6.2.1 Tree distances

By comparing trees of different patients, we may identify repeated patterns of tumor evolution. To do this in a principled way, we require a distance function $d(T, T')$ that quantifies the degree of differences between two trees T and T'. Many distance measures have been proposed for cancer phylogenies under the ISA [75, 76, 77, 78], including the parent-child distance (see Def. 2.2). To control for trees of varying sizes and mutation sets, we augment the parent-child distance to account for missing mutations in either tree and include a normalization factor (Fig. 6.2). This is formalized as follows.

Definition 6.1. The *normalized parent-child distance* $d_N(T, T')$ of two trees T and T' is the parent-child distance divided by twice the size of the vertex set $\Sigma = |V(T) \cup V(T')|$, i.e.

$$d_N(T, T') = \frac{|E(T) \triangle E(T')| + |V(T) \triangle V(T')|}{2\Sigma}. \tag{6.1}$$

6.2.2 Consensus tree problems

The problem of identifying repeated patterns of tumor evolution may be viewed as a consensus tree problem. The Single Consensus Tree (SCT) problem was posed and solved in a recent paper for trees with identical mutation sets using the parent-child distance.

Problem 6.1 (Single Consensus Tree (SCT) [75])**.** Given a set $\mathcal{T} = \{T_1, \cdots, T_n\}$ of trees on the same vertex set Σ, find a consensus tree R with vertex set Σ such that the total parent-child distance $\sum_{i=1}^{n} d(T_i, R)$ is minimum.

Representing evolutionary patterns common to a large number of patients by a single consensus tree is often too restrictive, as multiple subtypes with distinct evolutionary patterns and phenotypes exist even among cancers with the same primary location [176]. This limitation may be overcome by a natural extension of the SCT problem, where rather than

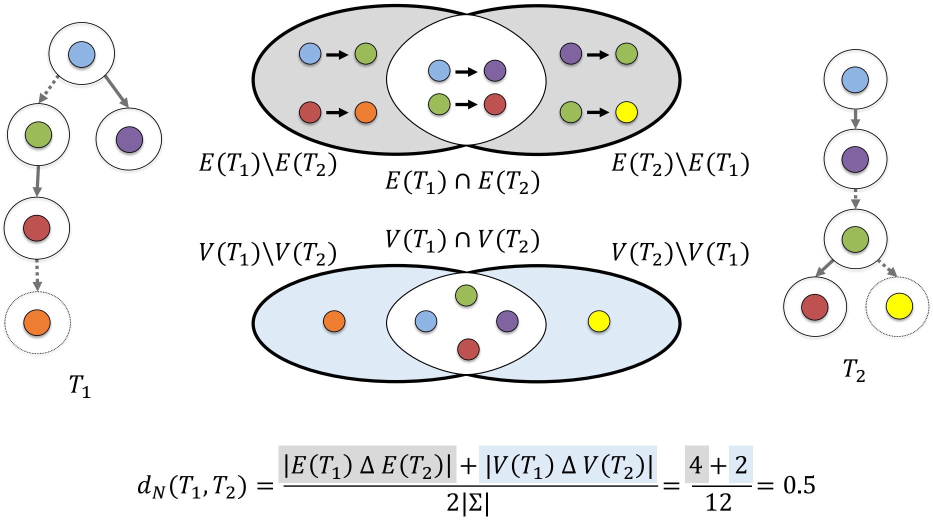


Figure 6.2: **Normalized parent-child distance accounts for varying mutation sets and tree sizes.** Here, Σ consists of six mutations (colored circles). The normalized parent-child distance $d_N(T_1, T_2) = 0.5$ of trees T_1 and T_2 is the sum of the sizes of the symmetric differences of their edge sets (light gray) and vertex sets (light blue) divided by $2|\Sigma|$.

finding a single consensus tree one simultaneously clusters patient trees and identifies a representative consensus tree for each cluster. In previous work, this was formalized as the Multiple Consensus Tree (MCT) problem [177].

Problem 6.2 (Multiple Consensus Tree (MCT) [177])**.** Given a set $\mathcal{T} = \{T_1, \cdots, T_n\}$ of trees with the same vertex set Σ and integer $k > 0$, find (i) a clustering $\sigma : [n] \to [k]$ of input trees into k clusters and (ii) a consensus tree R_j with vertex set Σ for each cluster $j \in [k]$ such that the total parent-child distance $\sum_{i=1}^{n} d(T_i, R_{\sigma(i)})$ is minimum.

There are three challenges that prevent the adoption of methods for the MCT problem to identify repeated evolutionary patterns. First, the application of phylogenetic techniques specialized for cancer sequencing data results in a large solution space $\mathcal{T}$ of plausible trees for each individual patient. Second, inference methods typically label vertices by mutation clusters rather than a single mutation. Recall that a tree T' is an *expansion* of a tree T if all mutation clusters of T have been expanded into ordered paths (see Fig. 6.4). Third, due to inter-tumor heterogeneity, the set of mutations across patients will vary, violating the constraint that patient trees are on the same set Σ of mutations.

Leveraging information across patients, we wish to resolve ambiguities in our input data and detect subtypes of evolutionary patterns by simultaneously (i) identifying a single expanded tree among the solution space of trees for each patient, (ii) assigning patients to clusters, and (iii) inferring a consensus tree summarizing the identified expanded trees for each cluster of patients. We formalize this as the MULTIPLE CHOICE CONSENSUS TREE problem (Fig. 6.1).

Problem 6.3. (MULTIPLE CHOICE CONSENSUS TREE (MCCT)) Given a family $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$ of sets of patient trees composed of subsets of mutations Σ and integer $k > 0$, find (i) a single tree $S_i \in \mathcal{T}_i$ for each patient $i \in [n]$, (ii) an expanded tree S'_i of each selected tree S_i, (iii) a clustering $\sigma : [n] \rightarrow [k]$ of patients into k (non-empty) clusters and (iv) a consensus tree R_j for each cluster $j \in [k]$ such that the total normalized parent-child distance $\sum_{i=1}^{n} d_N(S'_i, R_{\sigma(i)})$ is minimum.

The MCCT problem generalizes both the SCT and MCT problems when there are no mutation clusters and all patients have the same set of mutations. In particular, when there is only a single tree for each patient, the MCCT problem reduces to the MCT problem. For the case where, in addition to the previous, we seek only a single cluster ($k = 1$), the MCCT problem further reduces to the SCT problem.

6.2.3 Complexity

We start by noting that since the MCCT problem is a generalization for the MCT problem, any hardness result for MCT carries over to MCCT. Previously, the authors in [177] showed that MCT is NP-hard for the case where $k = O(n)$, which thus means that MCCT is NP-hard for the same case. Here, we prove a stronger result, showing that MCCT is NP-hard even when $k = 1$. Specifically, this section sketches a proof of NP-hardness for the MCCT problem by reducing from the canonical NP-hard problem of 3-SATISFIABILITY (3-SAT) [178]. The full proof can be found below in Section 6.6.

Theorem 6.1. MCCT is NP-hard even in the restricted case where (i) we seek a single consensus tree ($k = 1$), (ii) trees in $\mathcal{T}$ have the same vertex set Σ, and (iii) there are no mutation clusters.

Recall that in 3-SAT, we are given a Boolean formula $\phi = \wedge_{i=1}^{n}(y_{i,1} \vee y_{i,2} \vee y_{i,3})$ in 3-conjunctive normal form with m variables denoted by $\{x_1, \cdots, x_m\}$ and n clauses denoted by $\{c_1, \cdots, c_n\}$. We define $\gamma(y_{i,j}) = 1$ if literal $y_{i,j}$ is of the form x, and $\gamma(y_{i,j}) = 0$ if literal $y_{i,j}$ is of the form $\neg x$, where x is one of the variables. A truth assignment $\theta : [m] \rightarrow \{0, 1\}$ *satisfies* clause $c_i = (y_{i,1} \vee y_{i,2} \vee y_{i,3})$ if there exists a $j \in \{1, 2, 3\}$ such that $\theta(x) = \gamma(y_{i,j})$, where x is the variable corresponding to literal $y_{i,j}$. 3-SAT seeks to determine if there exists a truth assignment θ^* satisfying all clauses of ϕ.

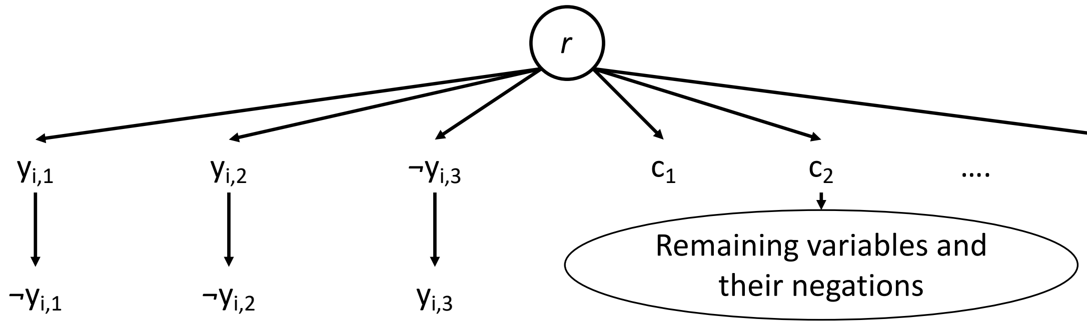


Figure 6.3: **An example of the gadget used in the NP-hardness proof for the MCCT problem.** This is just one the seven trees in collection $\mathcal{T}_i$ constructed from clause $c_i = y_{i,1} \vee y_{i,2} \vee y_{i,3}$ in our 3-SAT formula. This tree corresponds to the case where c_i is satisfied by both the first and second literal, but not the third.

Given an instance ϕ of 3-SAT, we reduce it to an MCCT instance $\mathcal{T}(\phi)$ as follows. To simplify the reduction, we assume that (i) ϕ has literals from three distinct variables within every clause, (ii) every variable and its negation appear in at least two clauses each, and (iii) a variable and its negation never appear in the same clause. These conditions are without loss of generality, as every ϕ that does not satisfy these conditions can be rewritten as an equisatisfiable formula ϕ' in polynomial time that adheres to the three conditions. We construct a family $\mathcal{T}(\phi) = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$ of sets of trees over the shared vertex/mutation set $\Sigma = \{r, x_1, \cdots, x_m, \neg x_1, \cdots, \neg x_m, c_1, \cdots, c_n\}$. Note that this shared vertex set contains a vertex for each positive and negative literal in ϕ, a vertex for every clause in ϕ, and an extra vertex r (i.e. $|\Sigma| = 2m + n + 1$).

For each clause $c_i = (y_{i,1} \vee y_{i,2} \vee y_{i,3})$ in ϕ, the family $\mathcal{T}(\phi)$ contains one set $\mathcal{T}_i$ comprised of seven trees. These trees correspond to the seven possible assignments of truth values to variables in c_i such that the clause is satisfied. Per our assumption that ϕ has clauses composed of distinct variables, there exist exactly seven distinct truth assignments that satisfy clause c_i. Consider one such assignment $\phi(x_1) = \gamma(y_{i,1})$, $\phi(x_2) = \gamma(y_{i,2})$, $\phi(x_3) \neq \gamma(y_{i,3})$, where x_1, x_2, x_3 are the variables corresponding to literals $y_{i,1}, y_{i,2}, y_{i,3}$, respectively. The tree representing this assignment in $\mathcal{T}_i$ is constructed as follows: (i) the tree has r as the root vertex; (ii) the root r has vertices $c_1, \cdots, c_n$ as children; (iii) the root also has children corresponding to each literal based on the assignment, i.e. $\{(r, y_{i,1}), (r, y_{i,2}), (r, \neg y_{i,3})\}$ for this example; (iv) each of these literals then has its negation as a child, i.e. $\{(y_{i,1}, \neg y_{i,1}), (y_{i,2}, \neg y_{i,2}), (\neg y_{i,3}, y_{i,3})\}$; (v) the remaining vertices (corresponding to variables and negations not in c_i) are added as children of the vertex labeled c_i. Note that r will always have $3 + n$ children corresponding to the three literals and n clauses. Fig. 6.3 shows an example.

This reduction can be performed in $O(|\mathcal{T}(\phi)| \cdot |\Sigma|) = O(n(2m + n + 1)) = O(n^2 + nm)$

time and is therefore polynomial. After constructing $\mathcal{T}(\phi)$, we can use an algorithm for MCCT to select one of the 7 trees from each set in $\mathcal{T}(\phi)$ in order to minimize the parent-child distance to a single consensus tree (i.e. $k = 1$). Note that minimizing the parent-child distance is equivalent to minimizing the normalized parent-child distance since all input trees have identical vertex sets and the same number of edges (i.e. the vertex symmetric difference in the numerator is zero, and the normalizing denominator is a constant scaling factor). Appendix B proves that ϕ has a satisfying assignment if and only if the optimal solution to this corresponding MCCT instance has a parent-child score of $2n(2m - 6)$. Moreover, we may use the consensus tree to recover a satisfying assignment for ϕ.

6.3 METHODS

In this section, we introduce Revealing Evolutionary Consensus Across Patients (RECAP), an algorithm to heuristically solve the Multiple Choice Consensus Tree (MCCT) problem. We first introduce a simplified version of the algorithm where all input trees from all patients are on the same mutation set and there are no mutation clusters (see Section 6.3.1). We then subsequently relax these requirements and show how we augment the algorithm to handle these two conditions (see Sections 6.3.2 and 6.3.3, respectively). Finally, Section 6.3.4 describes a model selection procedure for choosing k, the number of clusters.

6.3.1 Coordinate ascent heuristic for simple case

The MCCT problem models (i) the selection of one tree $S_i \in \mathcal{T}_i$ for each patient i, (ii) the surjective clustering function $\sigma : [n] \to [k]$ of the selected trees to one of k clusters, and (iii) the construction of multiple consensus trees $\{R_1, \ldots, R_k\}$ by minimizing the sum of normalized parent-child distances between consensus trees and the selected trees. To begin, we assume all trees from all patients have the same set of mutations and no clusters.

The pseudocode for our algorithm is given in Algorithm 6.1. We start by initializing a random selection of one tree for each patient. We also initialize a random assignment of patients to one of k clusters, ensuring that there is at least one patient per cluster. We then iterate between two steps: (i) finding an optimal consensus tree for the current selected trees assigned to each cluster, and (ii) selecting new trees for each patient and reassigning patients to clusters given the current consensus trees. We iterate between these two steps until convergence.

To perform step (i), we note that we can reduce this step into k independent instances of SCT, one for each cluster. The input to each SCT instance is simply the selected trees of

Algorithm 6.1: Coordinate Ascent Heuristic for Simple Case

```
Input: A collection $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_n\}$ of patients' tree sets and number $k > 0$ of
       clusters
Output: Selection of trees $\{S_1, \cdots, S_n\}$, consensus trees $\{R_1, \cdots, R_k\}$, and
        clustering $\sigma$ with smallest criterion score found.
$\{S_1, \cdots, S_n\} \leftarrow$ random tree selection for each patient $i$ from $\mathcal{T}_i$
$\sigma \leftarrow$ random surjective cluster mapping from $[n] \to [k]$
$\{R_1, \cdots, R_k\} \leftarrow$ Compute initial consensus tree for each cluster $j$ by running SCT
  algorithm on the set $\{S_i \mid \sigma(i) = j\}$.
$\Delta \leftarrow \infty, L \leftarrow \sum_{i=1}^{n} d(S_i, R_{\sigma(i)})$
while $\Delta > 0$ do
    for $j \leftarrow 1$ to $k$ do
        $R_j \leftarrow$ Update consensus tree for cluster $j$ by running SCT algorithm on the
          set $\{S_i | \sigma(i) = j\}$.
    end
    for $i \leftarrow 1$ to $n$ do
        $S_i, \sigma(i) \leftarrow$ Update selected tree and cluster for patient $i$ by directly
          computing $\text{argmin}_{T \in \mathcal{T}_i, j \in [k]} d(T, R_j)$
    end
    $\Delta \leftarrow L - \sum_{i=1}^{n} d(S_i, R_{\sigma(i)})$
    $L \leftarrow \sum_{i=1}^{n} d(S_i, R_{\sigma(i)})$
end
return $(\{S_1, \cdots, S_n\}, \{R_1, \cdots, R_k\}, \sigma)$
```

patients assigned to that cluster. The output is a consensus tree minimizing the parent-child distance to the input trees. Note that this is equivalent to minimizing the unnormalized parent-child distance; since we assume all patients have the same vertex set, the vertex symmetric diffrence in the numerator is equal to zero and normalization term in the parent-child distance function just becomes a constant scaling factor.

To perform step (ii), we iterate over all input trees for each patient. For each tree, we calculate the parent-child distance to the consensus tree for each cluster. We then select the tree and cluster that minimizes this distance for each patient.

While this algorithm is a heuristic, the total parent-child score is monotonically decreasing with each iteration. In step (i), the updated consensus tree is guaranteed to be optimal and so can only decrease the score. In step (ii), the tree selection and cluster assignment is only changed if it decreases the score. We restart the algorithm a user-specified number of times, each time with a different random initialization, and we return the solution with minimum parent-child distance across all restarts.

6.3.2 Varying mutation sets

We now adapt Algorithm 6.1 to be able to handle patients that have different sets of mutations. When patients in the input data have different mutation sets, some patients have many more mutations than other patients. When this occurs, minimizing the parent-child distance can often be achieved by putting the most massive trees alone in their own clusters with an identical consensus tree. To avoid this degenerate scenario, we introduce normalization to our distance function (see Definition 6.1).

On trees with identical vertex sets, optimizing this normalized distance simply reduces to optimizing the parent child distance, as we discussed above. However, with varying mutation sets, the numerator term containing the symmetric difference in vertex sets can no longer be assumed equal to zero. In most places in our algorithm, we can simply swap the distance function to normalized distances. However, this cannot immediately be done in step (i) since the SCT subroutine is designed to work on identical mutation sets and unnormalized distances.

To address this problem, we augment the input patient trees so that all augmented trees are on the same vertex set. As described in Section 6.2, all input trees share the same root vertex corresponding to the germline clone. We first add a new vertex labeled $\perp$ as a child of this shared root in all trees. For each patient tree, we then add new vertices for all mutations the tree is missing and attach each one as a child of $\perp$. We then run the algorithm as previously described on these augmented input trees. After the algorithm terminates, we post-process the consensus trees to remove the $\perp$ vertex along with all of its descendants, which we interpret as missing from this cluster.

The intuition behind this heuristic reduction is as follows. Consider a mutation b appearing in one tree but not the other. This mutation increases the vertex symmetric difference term in the normalized parent-child distance numerator. After augmenting the trees as described, this increase will now be captured by the symmetric difference in the edge sets of the augmented trees; the tree missing the mutation will now have the edge $(\perp, b)$, which is not contained in the other tree by construction.

6.3.3 Mutation clusters

In practice, patient input trees may have vertices that do not correspond to a single mutation, but in fact correspond to a set of mutations. We call vertices with multiple mutations mutation clusters. We interpret these clusters as implicitly representing another type of ambiguity in the patient trees where the linear ordering of mutations in the vertex is

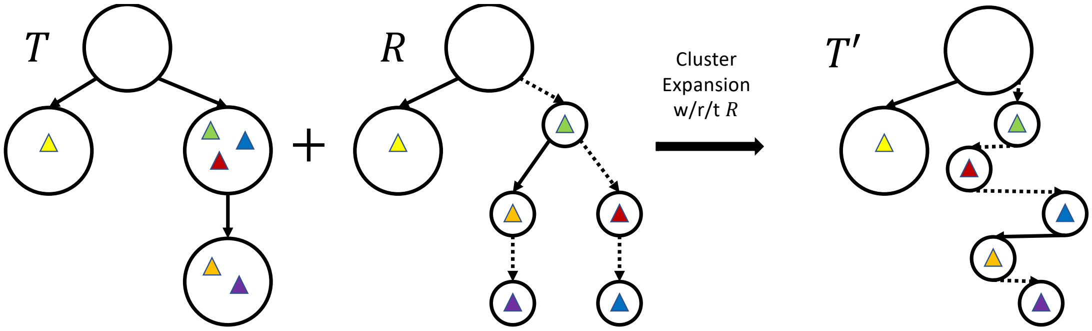


Figure 6.4: **An example of an optimal expansion of the mutation clusters of a tree T with respect to an expanded tree R.** Tree T contains mutation clusters, whereas tree R does not. Each mutation is denoted by a colored triangle. Matching edges between R and the expanded tree T' are denoted with a dashed line.

unknown. We wish to resolve all mutation clusters into a linear ordering of the mutations by leveraging information across patients. However, a naive expansion of all mutation clusters in all possible ways may dramatically increase the set of patient trees.

Solving the following optimization problem would allow us to resolve these clusters without explicitly enumerating all possible expansions. To start, we define an expansion of a mutation cluster just below (Fig. 6.4). Similarly, an *expanded tree* T' of T is obtained by expanding all mutation clusters of T into paths.

Definition 6.2. An *expansion* of a mutation cluster C is an ordered sequence $\Pi(C)$ of the mutations in C.

Problem 6.4. (Optimal Cluster Expansion (OCE)) Given a tree R with no mutation clusters and a tree T with at least one mutation cluster, find a tree T' such that (i) T' is an expansion of T, and (ii) T' minimizes the normalized parent-child distance to R out of all tree expansion of T.

We observe that when expanding mutation clusters, we cannot expand each mutation cluster in isolation since abutting clusters have edges that interact. Therefore, to solve this problem, we use dynamic programming (DP). The details of the polynomial time DP algorithm are given in Section 6.6. To incorporate support for mutation clusters into Algorithm 6.1, we run the DP subroutine on each patient tree considered in step (ii). This gives us the score of the best expansion of each tree in polynomial time, avoiding an exponential blow-up of the input tree set.

6.3.4 Model selection

In the above methodology, we gave the number k of clusters as an input to our algorithm. Clearly, the total normalized parent-child distance will decrease with increasing number k of clusters, with $k = n$ leading to a total distance of 0. Thus, we must choose the number of clusters necessary to explain the data without overfitting. Intuitively, what we seek is the the minimum number of clusters k, after which introducing additional clusters no longer leads to a meaningful decrease in our optimization criterion. In other words, this is the point at which the normalized parent-child distance "flattens". We capture this intuition with the following elbow approach.

Given an absolute threshold $t_a \geq 0$ and a percentage threshold $t_p \in (0, 1)$, we seek the largest k such that the following two conditions hold: (1) the change in the optimization criterion between $k - 1$ and k is greater than t_a, and (2) the percentage change in the optimization criterion between $k-1$ and k is greater than t_p. Selecting the largest k meeting these two conditions ensures that all larger k values must have a small marginal changes. The use of an absolute threshold just ensures that for normalized parent-child distances very close to 0, a fractional change to the total cost does not trigger the percentage change criterion. In practice, we set $t_a = 0.5$ and $t_p = 0.05$.

6.4 EVALUATION AND RESULTS

Section 6.4.1 compares RECAP to HINTRA [91] and REVOLVER [90] on simulated data, whereas Section 6.4.2 highlights the use of RECAP to identify repeated evolutionary trajectories in a non-small cell lung cohort [87] and a breast cancer cohort [179].

6.4.1 Simulations

We use simulations to evaluate our method. We generate three sets of simulation instances, with varying total number $|\Sigma|$ of mutations and number ℓ mutations per cluster. The first set has $|\Sigma| = \ell = 5$ mutations, the second set $|\Sigma| = 12$ total mutations and $\ell = 7$ mutations per cluster and the third set $|\Sigma| = \ell = 12$ mutations. For each set, we generate simulated instances with varying number $k^* \in \{1, 2, 3, 4, 5\}$ of clusters and number $n \in \{50, 100\}$ of patients, yielding an MCCT instance $\mathcal{T} = \{T_1, \ldots, T_n\}$ and solution $(\mathcal{R}^*, \Gamma^*, \sigma^*)$ as follows. First, we draw the patient clustering $\sigma^* : [n] \to [k]$ from a Dirichlet-multinomial distribution with concentration parameters $\alpha_1 = \ldots = \alpha_k = 10$ and the number of trials equal to the number of patients n. Next, for each cluster $j \in [k]$, we randomly pick ℓ

mutation without replacement from the set Σ, ensuring that mutation 0 is among the picked mutations. We then randomly generate a consensus tree R_j^* using Prüfer sequences [180], rooted at mutation 0. To obtain the set $\mathcal{T}_i$ of trees of patient $i \in [n]$, we simulate a bulk sequencing experiment by generating a matrix F of variant allele frequencies (with 5 bulk samples) obtained from mixing the vertices of the corresponding consensus tree $R_{\sigma(i)}$, and subsequently running SPRUCE [109]. For each simulation instance, parameterized by $|\Sigma|$, ℓ, n and k, we generate 20 instances. This amounts to a total of $3 \cdot 2 \cdot 5 \cdot 20 = 600$ instances.

We compare RECAP (50 restarts) to HINTRA [91] and REVOLVER [90]. We ran HINTRA using the following arguments:

```
$ ./Hintra-Lin -u <INPUT_FILE> <no_samples> <no_genes> 0.1 50 10
```

where 0.1 is the default of the discretization parameter, 50 is the default value for the number of EM restarts and 10 is the number of threads. We ran REVOLVER using the following default of the following R function:

```
revolver_fit(x, initial.solution = 1, max.iterations = 10, n = 10)
```

Fig. 6.6a shows that RECAP correctly selects the ground truth tree for each patient. REVOLVER, by contrast, only does so when the number k^* of simulated clusters equals 1 and performance decreases with increasing k^*. Indeed, in REVOLVER's model patient trees originate from a single generative model (which is a directed graph). This model assumption breaks down when there are distinct generative models, with varying sets of edges, for each patient cluster as is the case in our simulations. We were only able to run HINTRA for the $|\Sigma| = \ell = 5$ simulation instances, resulting in poor performance for varying number k^* of simulated clusters. Fig. 6.6b shows that RECAP's model selection criterion correctly identifies the simulated number k^* of clusters. REVOLVER's performance is slightly worse that RECAP, often overestimating the number of clusters. Next, we assess the accuracy of the patient clustering of RECAP and REVOLVER. Note that we did not include HINTRA in this analysis, as it is does not possess the capability to group patients into clusters with similar evolutionary trajectories. We find that RECAP correctly assigns pairs of patients to the same cluster (recall, Fig. 6.6c) and also correctly groups patients into distinct clusters (precision, Fig. 6.6d). We also report these results for each simulated dataset separately and observe similar trends; in particular, Fig. 6.7 shows simulation results for $|\Sigma| = \ell = 5$. Fig. 6.8 shows simulation results for $|\Sigma| = 12$ and $\ell = 7$. Fig. 6.9 shows simulation results for $|\Sigma| = \ell = 12$. Finally, we assess in Fig. 6.10 RECAP's stability with varying number of restarts, showing that RECAP quickly converges onto the ground truth solution.

In summary, our simulations demonstrate that RECAP outperforms existing methods, correctly reconstructing distinct evolutionary trajectories, selecting the correct tree per patient and correctly clustering patients together.

6.4.2 Real data

Non-small cell lung cancer cohort. We first run RECAP on the TRACERx dataset from [87], which contains whole-exome sequencing (500x depth) of tumors taken from patients ($n = 99$) with with non-small cell lung cancer. In the original study, phylogenetic trees were reconstructed for each patient with some patients having more than one proposed tree (median: 1 tree, maximum: 14 trees). The number of clones per patient ranges from 2 to 15. Furthermore, 85 patients have trees containing at least one mutation cluster, with a maximum mutation cluster size of 11. We additionally process these trees by restricting them to recurrent driver mutations, which we define to be mutations appearing in at least 10 patients. We run RECAP here with k ranging from 1 to 15 and with 5,000 restarts.

RECAP's model selection criterion identifies $k = 10$ distinct clusters (Fig. 6.11a). We note that as k increases, the clusters remain fairly stable in terms of the consensus trees found and the patient clustering, with each incremental cluster typically subdividing a previous cluster (Fig. 6.11b). The cluster size for the selected k ranges from a minimum of 4 patients to a maximum of 21 patients assigned to a particular cluster. Six of the consensus graphs we recover consist of at most one edge from germline to a driver mutation. The remaining four consensus trees have between two and three mutations.

We note that the authors in [90] likewise reported 10 distinct clusters for this dataset. Of these, the authors found five to have the strongest signal (C2, C3, C4, C6, C8). RECAP returns a consensus tree exactly matching two of these clusters, and very similar consensus trees for the remaining clusters. Moreover, the patients in these clusters are similarly clustered by RECAP.

We discuss Cluster 4 from RECAP, which we use as an illustrative example of how RECAP can use patterns observed in other patients to resolve ambiguities due to mutation clusters (Fig. 6.11c). The consensus tree for Cluster 4 contains an edge from germline to EGFR followed by an edge from EGFR to TP53 (matching cluster C4 in REVOLVER). We observe that in the input data, patient CRUK0015 has a single tree that after processing contains both of these edges, ordering EGFR and TP53 (Fig. 6.11d). As we would expect, patient CRUK0015 is assigned to Cluster 4. Moreover, this information then transfers via the consensus tree to resolve mutation clusters for 10 other patients in this cluster including CRUK0001, CRUK0004, CRUK0022, CRUK0024, CRUK0026, CRUK0048, CRUK0049, CRUK0051, CRUK0058, and CRUK0080. Indeed, it has been previously observed that EGFR and TP53 frequently co-occur, potentially having important clinical implications, and that in some patients EGFR proceeds TP53 [181]. To observe the remaining consensus trees in this cohort, see Fig. 6.12 and Fig. 6.13.

Breast cancer cohort. Razavi et al.[179] performed targeted sequencing of 1,918 tumors from 1,756 breast cancer samples, identifying copy number aberrations and single-nucleotide variants (SNVs) using a panel comprised of 468 genes. Here, we restrict our analysis to the subset of $n = 1,315$ patients with SNVs that occur in copy neutral autosomal regions. For each patient, we run SPRUCE [109] to enumerate all tumor phylogenies that explain the variant allele frequencies of the copy-neutral SNVs. Specifically, we identify between 1 to 6,332 trees per patient (median: 1). We further process these trees by restricting them to mutations that occur in at least 100 patients, yielding a set Σ of eight mutations. We run RECAP on this dataset with k ranging from 1 to 15 and with 1,000 restarts.

RECAP's identifies $k = 8$ distinct clusters for this dataset (Fig. 6.14a). Similarly to the lung cancer cohort, the clusters remain fairly stable in terms of the consensus trees found and the patient clustering (Fig. 6.14b). The cluster size for the selected k ranges from a minimum of 55 patients to a maximum of 410 patients assigned to a particular cluster. Two consensus trees have two mutations, the remaining six are comprised of a single mutation.

We focus our attention on Cluster 1, comprised of 71 patients. In particular, Patient P-0004859 has two input trees (Fig. 6.14c): TP53 and PIK3CA are children of MAP3K1 in tree T_1 while tree T_2 has a chain from MAKP1 to PIK3CA to TP53. As the consensus tree of this cluster has an edge from germline to EGFR and an edge from EGFR to TP53, RECAP selects tree T_2 for this patient (Fig. 6.14d). In turn, the consensus tree was informed by the mutation orderings of other patients, revealing shared evolutionary trajectories. In this way, the consensus tree facilitates the transfer of information across patients to resolve ambiguities in the solution space.

Previously, Khakabimamaghani et al. [91] used HINTRA to analyze this dataset, manually splitting the patients into four subtypes based on receptor status (HR+/HER2-, HR+/HER2+, HR-/HER2+ and Triple Negative). In the HR+/HER2- subtype, the authors found CDH1 commonly precedes PIK3CA. Without prior knowledge, RECAP recapitulates this finding in Cluster 7 with a consensus tree comprised of an edge from germline to CDH1 and then CDH1 to PIK3CA. When analyzing the 93 patients assigned to this cluster, we see that 87 patients ($\sim 93.5\%$) belong to the HR+/HER2- subtype. This finding demonstrates RECAP's ability to uncover cancer subtypes based on evolutionary trajectories. The remaining consensus trees found by RECAP can be found in Fig. 6.15 and Fig. 6.16.

6.5 DISCUSSION

In this paper, we formulated an optimization problem for simultaneously selecting a phylogeny for each patient in a cancer cohort, clustering these patients to account for subtype heterogeneity, and identifying a representative consensus tree for each patient cluster. After establishing the hardness of this problem, we proposed RECAP, a coordinate ascent algorithm as a heuristic for solving this problem. We included with this algorithm a way to handle patients with different sets of mutations as well as mutation clusters, something not previously handled in this line of work. The fact that our algorithm is capable of running over patients with different mutation clusters is particularly necessary in the whole-genome context, where the number of mutations necessitates clustering and there is variations in these clusters across patients. Moreover, we included a model selection criterion for identifying the number k of subtypes needed to explain a dataset. We validated our approach on simulated data, showing that RECAP outperforms existing methods that do not support diverse evolutionary trajectories. We demonstrated the use of RECAP on real data, identifying well-supported evolutionary trajectories in a non-small cell lung cancer cohort and a breast cancer cohort.

This work put forth a general framework for defining clusters of patients while reducing ambiguity inherent to the input data. We believe that this framework is adaptable and can be used to structure several avenues for future work. Broadly, these questions surround what makes two cancer phylogenies meaningfully similar and what are relevant underlying models that should be used to summarize shared evolutionary patterns. For instance, we currently support a variation on the parent-child distance to evaluate the difference between trees. However, there are other types of distance measures, such as the ancestor-descendent distance [75] or MLTD [76], that weigh discrepancies between trees differently. Exploring the trade-offs between distance metrics in more depth could lead to new insights. We currently require the consensus for each cluster to be a tree, but other graphical structures such as directed acyclic graph could be considered. This is especially useful when trying incorporate mutual exclusivity of driver mutations that occur in the same pathway into the inference. We could also consider incorporating auxiliary information, such as mutational signatures, into our model either via constraints or a secondary optimization criterion in order to test how clusters change when accounting for this incremental signal. Indeed, using mutational signatures as a constraint to improve the estimation of just a single patient tree has recently been done in [156]. On the theoretical side, we note that the current formulation is done using the infinite sites assumption. We hope to expand this work to the more comprehensive k-Dollo evolutionary model that allows for mutation losses [82]. Exploring such variations will not only shed light on solution space summarization, but will also shed light on the common evolutionary models generating the mutation patterns we observe in patient cohorts.

6.6 PROOFS

6.6.1 NP-Hardness Proof

Theorem 6.2. MCCT is NP-hard even in the restricted case where (i) we seek a single consensus tree ($k = 1$), (ii) trees in $\mathcal{T}$ have the same vertex set Σ, and (iii) there are no mutation clusters.

We begin our proof of the theorem by making several observations about the structure of any optimal consensus tree, regardless of patient tree selection. The first observation we make is about the objective function.

Observation 6.1. Let (Γ, R) be composed tree selection $\Gamma = \{S_1, \ldots, S_n\}$ and consensus tree R. Then, (Γ, R) achieves minimum normalized parent-child distance $d_N(\Gamma, R)$ if and only if (Γ, R) achieves minimum unnormalized parent-child distance $d(\Gamma, R)$.

Proof. Since the vertex sets of the selected trees Γ and the consensus tree R are identical, we have

$$d_N(\Gamma, R) = \sum_{i=1}^{n} \frac{|E(S_i) \triangle E(R)| + |V(S_i) \triangle V(R)|}{2|\Sigma|} \tag{6.2}$$

$$= \frac{1}{2|\Sigma|} \sum_{i=1}^{n} |E(S_i) \triangle E(R)| = \frac{1}{2|\Sigma|} \sum_{i=1}^{n} d(S_i, R) = \frac{1}{2|\Sigma|} d(\Gamma, R). \tag{6.3}$$

Thus, $d_N(\Gamma, R) \propto d(\Gamma, R)$.

QED.

Therefore, in the remainder of the proof, we will only consider unnormalized parent-child distances $d(\Gamma, R)$, which we refer to as simply ‘parent-child distance’. Next, because r is the root across all input trees, we show in the following lemma that if r is not the root of the consensus tree R, we can construct a new consensus tree R' with a smaller parent-child distance to any selection from $\mathcal{T}(\phi)$. This leads to a contradiction on the optimality of R.

Lemma 6.1. Any optimal consensus tree R with respect to $\mathcal{T}(\phi)$ has r as the root vertex.

Proof. Let $\Gamma = \{S_1, \cdots, S_n\}$ denote the trees selected from $\mathcal{T}(\phi)$ minimizing the total parent-child distance to R. Let v be the root vertex of R. Suppose for a contradiction that

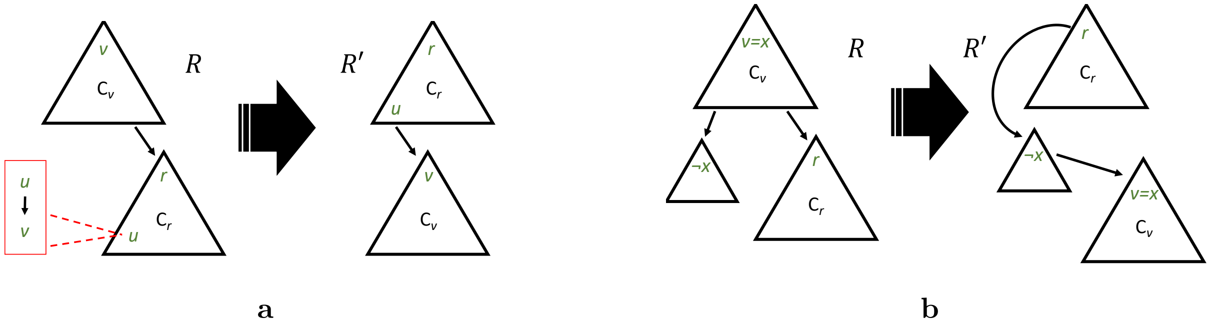


Figure 6.5: **The two cases considered in Lemma 6.1 when the root vertex v of R does not equal r.** (a) In the first case, subtree C_r of R contains an edge (u, v) that is present in at least one selected tree in Γ. (b) In the second case, no edge (u, v) of subtree C_r of R is present in at least one selected tree in Γ.

$v \neq r$. We will show how to construct a tree R' from R that contradicts the optimality of R, i.e. $d(\Gamma, R) > d(\Gamma, R')$. First, we remove the incoming edge to r, disconnecting R into two components: C_v containing v and C_r containing r. Initially, we set R' equal to C_r. When adding the remaining vertices from C_v to R', there are two cases to consider (Fig. 6.5).

1. The first case is when C_r contains some vertex u such that (u, v) is an edge in at least one selected tree in Γ. In this case, we reattach C_v rooted at v as a child of u in R'. Removing the incoming edge to r decreases the total distance to Γ by n since r has no parent in any tree of Γ by construction. Adding the edge (u, v) can increase the distance by at most $n - 1$, since (u, v) appears in at least one tree in Γ. Thus, the overall distance decreases, i.e. $d(\Gamma, R) > d(\Gamma, R')$, contradicting the optimality of R.

2. The second case is when C_r does not contain any vertices that appear as a parent to v in a selected tree in Γ. By construction, each clause vertex has r as its parent in every selected tree in Γ. Since r is contained in C_r, v cannot correspond to a clause vertex. It must thus correspond to some variable vertex $v = x$ that is never picked as a child of r in Γ. This implies that (i) x is a child of $\neg x$ in at least two trees in Γ, (ii) $\neg x$ must be a child of r in the same two trees in Γ, and (iii) $\neg x$ must also be in C_v. Note that (i) and (ii) are because we assume every literal appears in at least two clauses, and (iii) is because C_r does not contain a parent of v. To construct R', we perform two additional operations. First, we remove the incoming edge to $\neg x$ in C_v, and we reattach the resulting subtree rooted at $\neg x$ as a child of r in R'. Second, we take the remaining component of C_v rooted at v and make it a child of $\neg x$ in R' (i.e., we add

the edge $(\neg x, x)$).

We now show that R' has a smaller total distance to Γ than R. Similar to the above case, removing the incoming edge to r decreases the total distance by n. Adding the edge $(\neg x, x)$ increases the distance by at most $n-2$, since $(\neg x, x)$ appears in at least two trees in Γ. The final operation replaces the incoming edge to $\neg x$ with r. To see why the distance decrease, observe that $\neg x$ is a variable vertex. By construction the parent of $\neg x$ in any selected tree $S \in \Gamma$ is either (i) the negated variable vertex x, (ii) a clause variable, or (iii) the root r. By the premise, there is no selected tree $S \in \Gamma$ where x is the parent of $\neg x$. In addition, each clause variable can be at most once the parent of $\neg x$ among all selected trees Γ. Hence, r is the most frequent parent of $\neg x$ (occurring at least twice), and replacing the incoming edge to $\neg x$ with r also decreases the distance. Thus, we have contradicted the optimality of R implying that any optimal consensus tree must be rooted at r.

QED.

Since the edge (r, c_j) is in all input trees for all $j \in [n]$, these edges must also be present in any optimal consensus tree R. If these edges are not in the consensus tree, we again obtain a contradiction on the optimality of the consensus, as we show in the following lemma.

Lemma 6.2. Any optimal consensus tree R with respect to $\mathcal{T}(\phi)$ has the edge (r, c_j) for all $j \in [n]$.

Proof. Let $\Gamma = \{S_1, \cdots, S_n\}$ denote the trees selected from $\mathcal{T}(\phi)$ minimizing the total parent-child distance to R. Assume by way of contradiction that R does not contain the edge (r, c_j) for some j in $[n]$. We will construct a new consensus tree R' contradicting the optimality of R. The key fact to note is that r is the parent of c_j for every selected tree in $\Gamma = \{S_1, \cdots, S_n\}$ by construction. We consider two cases.

1. In the first case, the subtree rooted at c_j does not contain r. Let R' be obtained from R by regrafting the subtree rooted at c_j as a child of r. The parent-child distance will decrease by 2 for each selected tree in Γ, as the incoming edge to c_j now matches for all pairs of trees. Hence, $d(\Gamma, R) > d(\Gamma, R')$, contradicting the optimality of R.

2. In the second case, the subtree rooted at c_j contains r in R. This implies that r is not the root of R. By Lemma 6.1, this contradicts the optimality of R.

We get a contradiction in both cases. Thus, a solution of MCCT on $\mathcal{T}(\phi)$ has edges $\{(r, c_j)\}$ for all $j \in [n]$. QED.

We now proof a lower bound on the total parent-child distance of an optimal consensus tree R of the input trees $\mathcal{T}$. At a high level, this lower bound comes from the fact that edges of the form (c_i, x), for some clause vertex c_i and variable vertex x, only appear in the ith patient's set of input trees. There are $2m - 6$ such edges of this form in any selected patient tree $S_i \in \mathcal{T}_i$. We use this to identify $2(2m - 6)$ edges that must be in the symmetric difference between R and S_i. After doing this over all n patients and taking care to avoid double counting, we establish the lower bound of $2n(2m - 6)$.

Lemma 6.3. An optimal consensus tree R with respect to $\mathcal{T}(\phi)$ has a parent-child distance of at least $2n(2m - 6)$. Moreover, a consensus tree R achieving this lower bound cannot have edges of the form (c_i, x) or $(c_i, \neg x)$ for some clause c_i and variable x.

Proof. Let R be a consensus tree and $\Gamma = \{S_1, \cdots, S_n\}$ be a set of trees selected from $\mathcal{T}(\phi)$ minimizing the total parent-child distance $d(\Gamma, R)$. Let Δ be the multi-set composed of the symmetric differences $(E(S_i) \setminus E(R)) \cup (E(R) \setminus E(S_i))$ where $i \in [n]$. We claim that $|\Delta| \geq 2n(2m-6)$, which implies that $d(\Gamma, R) \geq 2n(2m-6)$. We will prove this constructively using the following algorithm.

```
$\Delta \leftarrow \emptyset$
for $i \leftarrow [n]$ do
    Let $X_i$ be the set of literals corresponding to variables absent in $c_i$
    for $x \leftarrow X_i$ do
        if$(c_i, x) \notin E(R)$ then
            Let $(v, x)$ be the unique incoming edge to $x$ in $R$
            $\Delta \leftarrow \Delta \cup \{(c_i, x) \in E(S_i), (v, x) \in E(R)\}$
        else
            Let $c_j$ be a clause containing $x$ and let $(v, x)$ be the
            unique incoming edge to $x$ in $S_j$
            $\Delta \leftarrow \Delta \cup \{(v, x) \in E(S_j), (c_i, x) \in E(R)\}$
```

We claim that at iteration i, we have identified $2i(2m - 6)$ edges of Δ, which we prove by induction on i.

- *Base case* $i = 1$: Consider $S_1 \in \Gamma$ corresponding to clause c_1. Let X_1 be the set of literals corresponding to variables that are absent in c_1. Since the literals from each clause in ϕ come from three distinct variables, there are a total of $|X_1| = 2m-6$ literals corresponding to variables absent in c_1. Thus, there are a total of $|X| = 2m - 6$ edges (c_1, x) in tree S_1 where $x \in X_1$.

For each such edge (c_1, x) in S_1, there are two cases. If (c_1, x) is not in R, then (c_1, x) in S_1 increases the symmetric difference by 1. Furthermore, since x is not the root of R (by Lemma 6.1), the incoming edge to x in R is missing from S_1 (i.e., (v, x) for some $v \in V$). Altogether, this increased the symmetric difference by 2.

Now consider the case where (c_1, x) is in R. By construction, (c_1, x) is not in any other input tree in $\mathcal{T}(\phi) \setminus \{\mathcal{T}_1(\phi)\}$ and is thus also absent from $\{S_2, \ldots, S_n\}$. However, we must charge this edge carefully in order to not double count edges across selected trees in future steps. By our restrictions on ϕ, x appears in some clause, c_j, for some j in $[n]$. By construction, the corresponding selected tree S_j must then contain the edge (v, x) where $v \in \{r, \neg x\}$. Either way, x has a different parent $v \neq c_1$ in S_j compared to R and we have identified two edges, (c_1, x) and (v, x) of Δ.

Repeating this process for all $2m - 6$ edges in S_1 where c_1 is the parent, we find $2(2m - 6)$ distinct edges to add to the symmetric difference.

- *Inductive step $i > 1$*: By the inductive hypothesis, we assume we are able to identify $2(i-1)(2m-6)$ distinct edges of Δ. Now consider the ith selected tree in our ordering, S_i. We claim that we can identify an additional $2(2m - 6)$ distinct edges of Δ. Let X_i be the set of literals corresponding to variables that are absent in c_i. As before, the selected tree S_i contains $2m - 6$ edges (c_i, x) where $x \in X_i$.

  For each such edge (c_i, x) in S_i, we again distinguish two cases. If (c_i, x) is not in R, then R and S_i must have different parents for x. Indeed, as described in the base case, (c_i, x) is present in S_i but not in R and conversely the incoming edge (v, x) to x in R is also missing from S_i (which exists, as x cannot be the root of R by Lemma 6.1). Hence, the edge (c_i, x) of S_i and the (v, x) of R are edges of Δ. We now need to show that the edge (c_i, x) of S_i was not added in a previous iteration to Δ. To see why this is not the case, observe that only two types of edges from Γ were previously added. The first type are edges (c_j, x) that were added in step $j < i$. The second type are edges (v, x) of a tree S_l where $v \in \{r, \neg x\}$, corresponding to a variable x that is *present* in some clause c_l. Both cases do not apply, as $j < i$ and $c_i \notin \{r, \neg x\}$. Hence, these two edges of Δ were not previously considered.

  The second case is when (c_i, x) is in R. Let c_l be a clause containing x (which must exist by definition of ϕ). By construction, the corresponding selected tree S_l contains the edge (v, x) where $v \in \{r, \neg x\}$. Since $c_i \neq v$, clearly the edge (c_i, x) of R and the edge (v, x) of S_l are present in Δ. We claim that the edge (v, x) of S_l was not added to Δ in a previous iteration $j < i$. Inspection of the algorithm reveals that this edge

(v, x) can only be added in a previous iteration j if (c_j, x) is an edge of R. However, by our premise, R already contains the edge (c_i, x). Thus, since R is a tree, there is no edge (c_j, x) in R where $c_j \neq c_i$. Hence, the edge (v, x) of Δ was not previously considered.

Repeating this process for all $2(2m - 6)$ edges in S_i where c_i is the parent, we find an additional $2(2m-6)$ distinct edges to add to the symmetric difference. Combining this total with the inductive hypothesis we have identified $2i(2m-6)$ distinct edges of Δ upon completion of iteration i.

Thus, when the algorithm terminates at iteration n, we have identified $2n(2m - 6)$ distinct edges of Δ. To prove the final part of this lemma, we note that each literal x appears in at least two clauses by our assumption on ϕ. In the case where (c_i, x) is in R, we can therefore find two edges to add to the symmetric difference for *every* selected tree S_j corresponding to a clause c_j containing x (i.e., at least 4 edges will be added). In this case, R fails to achieve the lower bound of $2n(2m - 6)$. QED.

The following lemma again follows from a proof by contradiction on the optimality of R if this were not the case.

Lemma 6.4. Let consensus tree R and selected trees $\Gamma = \{S_1, \ldots, S_n\}$ be an optimal solution to MCCT instance $\mathcal{T}(\phi)$. If $d(\Gamma, R) = 2n(2m - 6)$ then either $\{(r, x), (x, \neg x)\} \subseteq E(R)$ or $\{(r, \neg x), (\neg x, x)\} \subseteq E(R)$ for all variables x. Moreover, whichever set appears in $E(R)$ must also occur in every tree of Γ_x, where Γ_x denotes the subset of selected trees Γ corresponding to clauses containing x or $\neg x$.

Proof. Let consensus tree R and selected trees $\Gamma = \{S_1, \ldots, S_n\}$ be an optimal solution to MCCT instance $\mathcal{T}(\phi)$ with total distance $d(\Gamma, R) = 2n(2m-6)$. Suppose by way of contradiction that there exists a variable x such that neither $\{(r, x), (x, \neg x)\}$ nor $\{(r, \neg x), (\neg x, x)\}$ appear in $E(R)$. We consider three cases for the arrangement of x and $\neg x$ in R.

(i) Vertex x or $\neg x$ is the root of R:

This case contradicts the optimality of R by Lemma 6.1.

(ii) Vertex x or $\neg x$ has a different variable vertex as a parent:

Let $y \neq \neg x$ be a variable vertex that is parent of x (the case for $\neg x$ is symmetric). As this arrangement never appears in any input tree in $\mathcal{T}(\phi)$, we have the edge (y, x) of R incurs a cost of 2 in each selected tree in Γ. Thus, it is straightforward to construct a tree R' with a lower distance to Γ. That is, consider a selected tree $S \in \Gamma_x$. If S

contains the edge (r, x) then move x to be the child of r in R'. Otherwise, if S contains the edge $(\neg x, x)$ then move x to be the child of $\neg x$ in R'. In both cases, the total distance $d(\Gamma, R')$ is strictly smaller than the original distance $d(\Gamma, R)$ (by at least a value of 2), leading to a contradiction.

(iii) Vertex x or $\neg x$ has a clause variable as a parent:

This contradicts R achieving the lower bound distance $d(\Gamma, R) = 2n(2m - 6)$ by Lemma 6.3.

(iv) Both x and $\neg x$ have r as a parent:

No clause variable c_i for $i \in [n]$ has a child in R by Lemma 6.3. We can obtain the lower bound distance of $2n(2m-6)$ by counting edges of the form $(c_i, x) \subset E(S_i)$ across all S_i that must be in the multi-set Δ composed of symmetric differences. For each such edge (c_i, x), a counterpart edge $(p, x) \subset E(R)$ for some $p \in V$ must also be in Δ. We now need to find one more edge in Δ to obtain a contradiction. By construction, Γ_x cannot be empty and each tree in Γ_x must either contain $\{(r, x), (x, \neg x)\}$ or $\{(r, \neg x), (\neg x, x)\}$. WLOG assume $S \in \Gamma_x$ contains $\{(r, x), (x, \neg x)\}$. Then, $(x, \neg x) \subset E(S)$ must also be in δ, contradicting the fact that R achieves the lower bound.

Thus, either $\{(r, x), (x, \neg x)\} \subseteq E(R)$ or $\{(r, \neg x), (\neg x, x)\} \subseteq E(R)$ for all variables x. To prove the final point, WLOG assume $\{(r, x), (x, \neg x)\} \subseteq E(R)$; if there exists a tree in $S \in \Gamma_x$ such that $\{(r, \neg x), (\neg x, x)\} \subseteq E(S)$, both of these edges must also exist in Δ implying that R does not achieve the lower bound.

QED.

In one direction, we now show that given an optimal consensus tree with parent-child distance $2n(2m - 6)$, we can read off the satisfying assignment by looking at the children of the root vertex r in R. In the other direction, we use a satisfying assignment to identify which tree we should select for each patient (i.e. the one corresponding to a satisfying assignment) and build the consensus tree R, where the satisfied literals hang off of the root.

Lemma 6.5. A Boolean formula ϕ meeting our three restrictions is satisfiable if and only if $\mathcal{T}(\phi)$ has an optimal single consensus tree with parent-child distance $2n(2m - 6)$.

Proof. $(\Rightarrow)$ Let ϕ be satisfiable. We will directly construct a solution to the corresponding MCCT problem achieving the lower bound. Let θ be a satisfying assignment. Consider an arbitrary clause c_i containing variables x_1, x_2, x_3. We select tree $S_i \in \mathcal{T}_i$ such that $(r, x_j) \in E(S_i)$ if $\theta(x_j) = 1$ and $(r, \neg x_j) \in E(S_i)$ if $\theta(x_j) = 0$. Note that such a tree must exist in $\mathcal{T}_i$ by construction. We construct a consensus tree R with root r and edges (r, c_i) for all $i \in [n]$. We then add edges $\{(r,x)(x,\neg x\}$ if $\theta(x_j) = 0$ or edges $\{(r,\neg x)(\neg x, x\}$ if $\theta(x_j) = 1$. Finally, observe that $d(S_i, R)$ is equal to $2(2m-6)$, due to edges of the form $(c_i, x) \in E(S_i)$ and the corresponding edge $(v, x) \in E(R)$ where $v \in \{r, \neg x\}$. Since i was arbitrary, we constructed consensus tree R with distance $2n(2m-6)$ across all selected trees. By Lemma 6.3, R must be optimal.

($\Leftarrow$) Let $\mathcal{T}(\phi)$ have an optimal consensus tree R and tree selection Γ with distance $d(\Gamma, R) = 2n(2m-6)$. By Lemma 6.4, R must have either the edge (r,x) or $(r,\neg x)$ for all variables x (but not both). Consider the assignment θ which sets a variable x equal to 1 if literal x is a child of the root and equal to 0 if literal $\neg x$ is a child of the root in R. We claim θ is a satisfying assignment for ϕ. Let c_i be an arbitrary clause in ϕ, and let $S_i \in \mathcal{T}_i$ be the tree selected by MCCT. For each variable x in clause c_i, either $(r,x) \in E(S_i)$ if $\theta(x) = 1$ or $(r,\neg x) \in E(S_i)$ if $\theta(x) = 0$ by Lemma 6.4. By construction of trees in $\mathcal{T}_i$, this implies that $\theta(x_j) = \gamma(y_i, j)$ for $j \in [3]$; thus, θ corresponds to a satisfying assignment for clause c_i. Since c_i was arbitrary, all clauses in ϕ must be satisfied. QED.

This then concludes the proof of the theorem as we have now shown that MCCT is hard even for this special case where $k = 1$ on identical patient mutation sets with no mutation clusters by a polynomial reduction from 3-SAT; thus, MCCT is NP-hard in general.

6.6.2 Dynamic programming algorithm for expanding mutation clusters

Here we describe the dynamic programming (DP) algorithm we use to solve the OCE problem. Recall that in this problem, we are given a tree R with no mutation clusters and a tree T with at least one mutation cluster. We wish to find a tree T' such that (i) T' is an expansion of T, and (ii) T' minimizes the normalized parent-child distance to R out of all tree expansion of T.

In the following recursion of our DP, each subproblem is an expansion of a subtree of T. For each subtree, we look at all possible start and end mutations for the expansion of the root mutation cluster; intuitively, these are the mutations that interact with mutations outside of the cluster. Let T_C be the subtree of T rooted at the vertex corresponding to mutation cluster C and let $\mu(C)$ be the set of mutations in C. Given R and T as defined above, we define the function $f(C, s, t)$ to be the maximum number of matching pairs of edges between R and any expansion of the subtree T_C, such that the mutation cluster $C \in V(T)$

is expanded into a path starting with mutation s and ending with mutation t.

$$f(C,s,t) = \begin{cases} -\infty, & \text{if } |\mu(C)| > 1 \text{ and } s = t, \\ 0, & \text{if } |\mu(C)| = 1 \text{ and } C \text{ is a leaf,} \\ g(C,s,t) + h(C,s,t), & \text{otherwise,} \end{cases} \tag{6.4}$$

where we have

$$h(C,s,t) = \sum_{W \in \delta(C)} \max_{s',t' \in \mu(W)} \left\{ \mathbb{1}((t,s') \in E(R)) + f(W,s',t') \right\}, \tag{6.5}$$

which recursively finds the best scoring expansion for the children $\delta(C)$ of C given t, adding an additional match if there is an edge between t and the expansion of a child in $E(R)$. In the recurrence, we have $g(C,s,t)$, which is defined as the maximum number of matching pairs of edges between R and any expansion of the mutation cluster C starting with mutation s and ending with mutation t. If $s = t$, this is defined to be zero. For now, assume we have oracle access to this value. We will give an explicit algorithm for calculating $g(C,s,t)$ later in this section.

Theorem 6.3. Given a rooted tree R with no mutation clusters and a rooted tree T with at least one mutation cluster, taking $OCE(T,R) = \max_{s,t \in \mu(r(T))} f(r(T),s,t)$ finds the optimal value for the OCE problem.

Proof. We prove this theorem by induction on $\ell \in [|V(T)|]$, where ℓ denotes the index of a vertex within a topological ordering of $V(T)$ such that each child vertex comes prior to its parent.

Base case: When $\ell = 0$, we have that the vertex C_0 must be a leaf. Let $s,t \in \mu(C_0)$ be arbitrary. There are three cases to consider:

1. If C_0 is not a mutation cluster (i.e., $|\mu(C_0)| = 1$, $s = t$), then there are no edges in the expansion of T_{C_0} since this is a leaf. Hence, $f(C_0,s,t) = 0$ as claimed.

2. Else if C_0 is a mutation cluster and $s = t$, then by definition of f the mutation s needs to be repeated twice, once at the start and once at the end of the expansion. Since this degenerate case is not allowed, f correctly returns $-\infty$ so it is never selected. Indeed, any pair of distinct mutations from C_0 will achieve a better score.

3. Finally, if C_0 is a mutation cluster and $s \neq t$, $f(C,s,t) = g(C_0,s,t)$ since $\delta(C_0) = \emptyset$, i.e., C_0 has no children. This is again correct by the definition of $g(C_0,s,t)$.

Inductive step: Assume the recursion is correct for all subtrees up to the one rooted at $C_{\ell-1}$. We now consider the subtree rooted at C_ℓ. There are again three cases to consider:

1. If C_ℓ is a leaf, then the correctness follows from the same analysis as in the base case.

2. Else if C_ℓ is a mutation cluster and $s = t$, then this degenerate case is not allowed. As was shown in the base case, f correctly returns $-\infty$ so $s = t$ is never selected.

3. Finally, if C_ℓ is a mutation cluster and $s \neq t$, the score of an expansion of T_{C_ℓ} is simply the sum of (i) the number of edges from an expansion of C_ℓ starting with s and ending with t that exist in $E(R)$, (ii) the number of edges between t and the root of the expansion of each child in $E(R)$, and (iii) the score of the expansion of the subtree at each child. We see that $g(\cdot,\cdot,\cdot)$ maximizes (i) while $h(\cdot,\cdot,\cdot)$ jointly maximizes (ii) and (iii).

It then follows that $\max_{s,t\in\mu(r(T))} f(r(T), s, t)$ is the maximum number of matching pairs of edges between R and any expansion of T, since we exhaust all starting and ending points in the expansion of the root. Let T' be an expansion of T that achieves this score. This implies that T' maximizes $|E(T') \cap E(R)|$, which in turn implies that it minimizes the parent-child distance $d(T', R)$ out of all expansions of T. Note that since all expansions of T have the same number of edges, T' also minimizes $d_N(T', R)$ and is an optimal solution to the OCE problem. QED.

Next, we show how we can efficiently calculate $g(C, s, t)$ with respect to tree R. The pseudocode is given in Algorithm 6.2. The intuitive idea is we wish to identify all edges in $E(R)$ that can be preserved when expanding mutation cluster C. An upper bound on this number is of course the edges in $E(R)$ with both endpoints in $\mu(C)$. We call this restricted subgraph $R_{\mu(C)}$.

Definition 6.3. The graph R *restricted* to vertex set Σ, denoted R_Σ, is a directed graph on vertex set Σ with edges $\{(u, v) \in E(R) | u \in \Sigma, v \in \Sigma\}$.

However, it is not always possible to maintain all of these edges since R is a tree, and we expand C into a path. Therefore, we first identify the connected components of $R_{\mu(C)}$. To decompose these components into paths we perform the following operations:

1. If there is a path from s and t in some component, we carefully select an edge to break along this path since all the mutations will eventually need to be added somewhere between s and t. In particular, we break the edge with the highest degree parent along the path.

2. We also break any incoming edge to s or outgoing edge from t since these must be the start and endpoints of the expansion, respectively.

3. We then break along edges where a parent has more than one child to remove any remaining branches.

Finally, we stitch the resulting paths back together into one long path $\Pi(C)$, which we treat as the expansion of C. We ensure that this expansion starts with s and ends with t, but otherwise the ordering does not matter.

Algorithm 6.2: Single Cluster Expansion

```
Input: A rooted tree $R$ with no mutation clusters, a rooted tree $T$ with at least one
       mutation cluster, a mutation cluster $C \in V(T)$, and mutations $s, t \in \mu(C)$.
Output: The maximum number of edges shared with $E(R)$ in an expansion of $C$
        starting with $s$ and ending with $t$.
if $|\mu(C)| > 1$ and $s = t$ then
    Raise Error
end
if $|\mu(C)| = 1$ then
    return 0
end
$\Theta \leftarrow$ Construct $R_{\mu(C)}$. Store resulting connected components.
if $\exists$ an edge $(u, s)$ within a component $\ell \in \Theta$ then
    Split $\ell$ into two components by removing $(u, s)$. Update $\Theta$.
end
if $\exists$ an edge $(t, v)$ within a component $\ell \in \Theta$ then
    Split $\ell$ into two components by removing $(t, v)$. Update $\Theta$.
end
if $\exists$ a path $p$ from $s$ to $t$ within a component $\ell \in \Theta$ then
    Find vertex $u$ on $p$ with highest degree.
    Split $\ell$ into two components by removing the unique edge $(u, v)$ such that $v$ is on
      path $p$. Update $\Theta$.
end
while $\exists \ell \in \Theta$, $v \in \ell$ such that $v$ has more than one child do
    Remove edges to all but one child of $v$. Update $\Theta$.
end
$\Pi(C) \leftarrow \ell \in \Theta$ such that $s \in \ell$.
for $\ell \in \Theta$ such that $s, t \notin \ell$ do
    Append component $\ell$ to path $\Pi(C)$ (i.e., $\Pi(C) \leftarrow \Pi(C) + \ell$).
end
$\Pi(C) \leftarrow \ell \in \Theta$ such that $t \in \ell$.
return $|E(\Pi(C)) \cap E(R)|$
```

Theorem 6.4. Algorithm 6.2 finds the maximum number of matching pairs of edges between R and any expansion of the mutation cluster C starting with mutation s and ending with mutation t.

We prove this theorem by first establishing an upper bound on this distance and then showing our algorithm achieves the upper bound.

Lemma 6.6. The maximum number of matching pairs of edges between R and any expansion $\Pi(C)$ of the mutation cluster C starting with mutation s and ending with mutation t is less than or equal to:

$$|E(R_{\mu(C)})| - \mathbb{1}(s) - \mathbb{1}(t) - \mathbb{1}(s,t) - \sum_{v \in V(R_{\mu(C)})} (|\delta(v)| - 1) \tag{6.6}$$

where $\mathbb{1}(s)$ indicates if s has a parent of outdegree 1 in $R_{\mu(C)}$, $\mathbb{1}(t)$ indicates if t has a child in $R_{\mu(C)}$, and $\mathbb{1}(s,t)$ indicates if there is a path from s to t in $R_{\mu(C)}$ without a vertex having more than one child.

A sketch for the proof of this lemma is the following. We can only keep one edge per parent vertex in $R_{\mu(C)}$ when we build the expansion path. We also must break any incoming edge to s and any outgoing edge from t. Finally, we also need to break an edge on the path from s to t to ensure that s is the starting vertex and t is the ending vertex, with all other vertices spliced between them. We want to make sure we do not double count broken edges, which results in the use of indicator variables.

Lemma 6.7. Algorithm 6.2 finds

$$|E(R_{\mu(C)}| - \mathbb{1}(s) - \mathbb{1}(t) - \mathbb{1}(s,t) - \sum_{v \in V(R_{\mu(C)})} (|\delta(v)| - 1) \tag{6.7}$$

matching pairs of edges between R and any expansion of the mutation cluster C starting with mutation s and ending with mutation t.

The proof of this lemma follows by counting the number of edges broken by Algorithm 6.2. Finally, we observe that the running time of Algorithm 6.2 is $O(|\Sigma|^5)$. To see this, note that there are $|V(T)| = O(|\Sigma|)$ possible roots $C \in V(T)$, each with $|\mu(C)|^2 = O(|\Sigma|^2)$ start and endpoints implying there are $O(|\Sigma|^3)$ distinct subproblem of $f(\cdot,\cdot,\cdot)$. Each subproblem then requires $O(|\Sigma|)$ time to compute $g(\cdot,\cdot,\cdot)$ and $h(\cdot,\cdot,\cdot)$. The former requires $O(\Sigma)$ time since it is comprised of a constant number of graph traversals and the latter requires $O(\Sigma^2)$ time

Algorithm 6.3: Generalized Coordinate Ascent Heuristic

```
Input: A collection $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_n\}$ of patients' tree sets and number $k > 0$ of
       clusters
Output: Selection of trees $\{S_1, \cdots, S_n\}$, consensus trees $\{R_1, \cdots, R_k\}$, and
        clustering $\sigma$ with smallest criterion score found.
$\mathcal{T}' \leftarrow$ augment the tree for each patient $i$ from $\mathcal{T}_i$ to span all mutations plut $\perp$
$\{S_1, \cdots, S_n\} \leftarrow$ random tree selection for each patient $i$ from $\mathcal{T}_i'$
$\sigma \leftarrow$ random surjective cluster mapping from $[n] \to [k]$
$\{R_1, \cdots, R_k\} \leftarrow$ Compute initial consensus tree for each cluster $j$ by running SCT
  algorithm on the set $\{S_i | \sigma(i) = j\}$.
$\Delta \leftarrow \infty, L \leftarrow \sum_{i=1}^{n} OCE(S_i, R_{\sigma(i)})$
while $\Delta > 0$ do
    for $j \leftarrow 1$ to $k$ do
        $R_j \leftarrow$ Update consensus tree for cluster $j$ by running SCT algorithm on the
          set $\{S_i | \sigma(i) = j\}$.
    end
    for $i \leftarrow 1$ to $n$ do
        $S_i, \sigma(i) \leftarrow$ Update selected tree and cluster for patient $i$ by directly
          computing $\text{argmin}_{T \in \mathcal{T}_i', j \in [k]} OCE(T, R_j)$
    end
    $\Delta \leftarrow L - \sum_{i=1}^{n} OCE(S_i, R_{\sigma(i)})$
    $L \leftarrow \sum_{i=1}^{n} OCE(S_i, R_{\sigma(i)})$
end
Remove $\perp$ and all its descendants for all trees in the selected set $\{S_1, \cdots, S_n\}$ and
  consensus set $\{R_1, \cdots, R_k\}$
return $(\{S_1, \cdots, S_n\}, \{R_1, \cdots, R_k\}, \sigma)$
```

when we carefully account for the number of times a child is recursed on:

$$O\left(\sum_{C \in V(T)} \sum_{s,t \in \mu(C)} \left[|\Sigma| + \sum_{W \in \delta(C)} \sum_{s',t' \in \mu(W)} 1\right]\right) \tag{6.8}$$

$$= O\left(|\Sigma|^4 + \sum_{C \in V(T)} \sum_{s,t \in \mu(C)} \sum_{W \in \delta(C)} \sum_{s',t' \in \mu(W)} 1\right) \tag{6.9}$$

$$= O\left(|\Sigma|^4 + \sum_{C \in V(T)} \sum_{W \in \delta(C)} \sum_{s,t \in \mu(C)} \Sigma^2\right) \tag{6.10}$$

$$= O\left(|\Sigma|^4 + \sum_{C \in V(T)} \sum_{W \in \delta(C)} \Sigma^2 \cdot \Sigma^2\right) \tag{6.11}$$

$$= O(\Sigma^5) \tag{6.12}$$

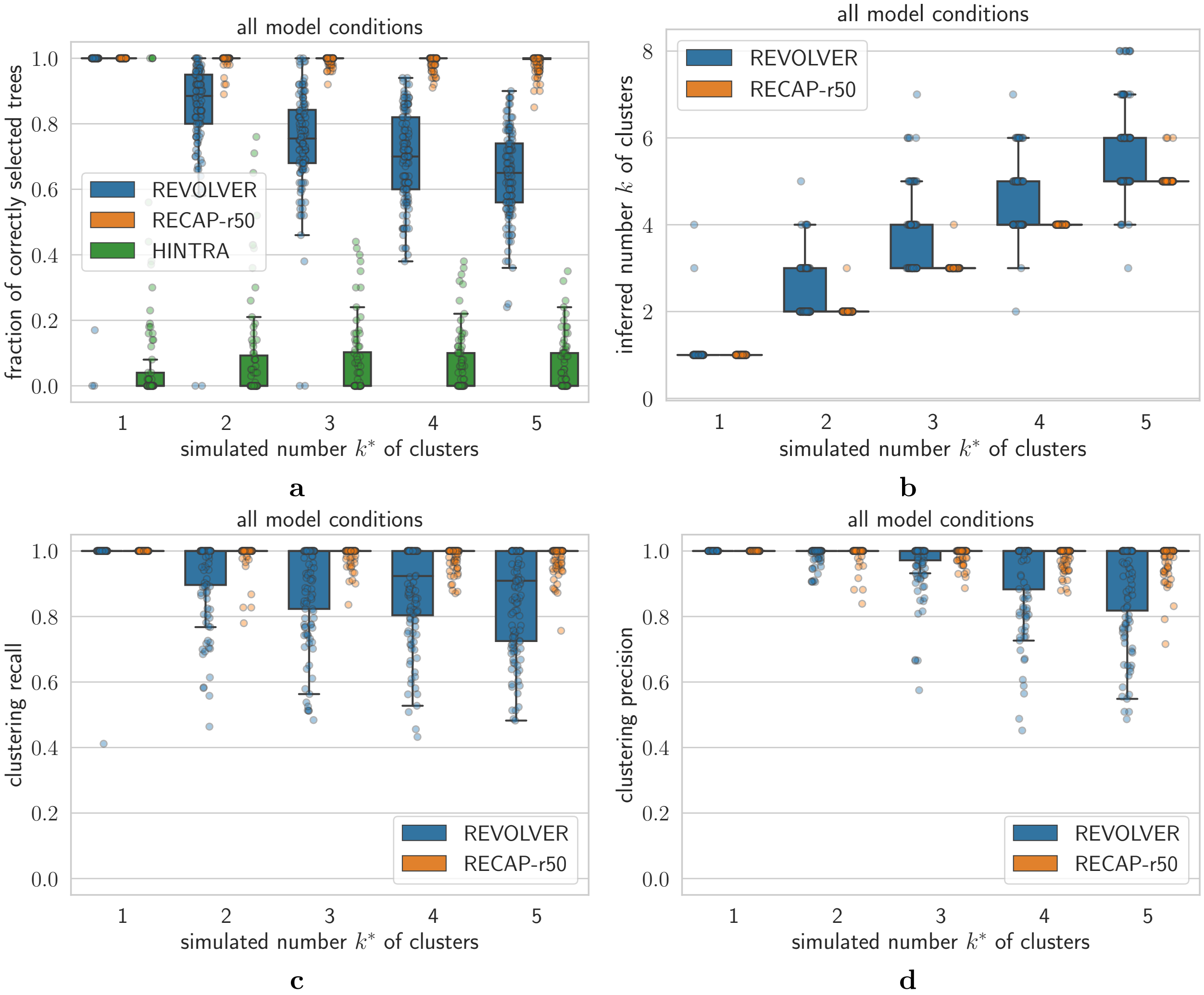


Figure 6.6: **Simulations show that RECAP accurately solves the MCCT problem, outperforming HINTRA [91] and REVOLVER [90].** Results for all simulation conditions. (a) The fraction of patients with correctly inferred trees by each method. (b) The number k of patient clusters inferred by each method. (c) The fraction of patient pairs that are correctly clustered together. (d) The fraction of patient pairs that are correctly put in separate clusters. Panel (a) shows only $|\Sigma| = \ell = 5$ results for HINTRA, due to scaling issues. No results are shown in (b)-(d) for HINTRA, as this method does not infer clusters.

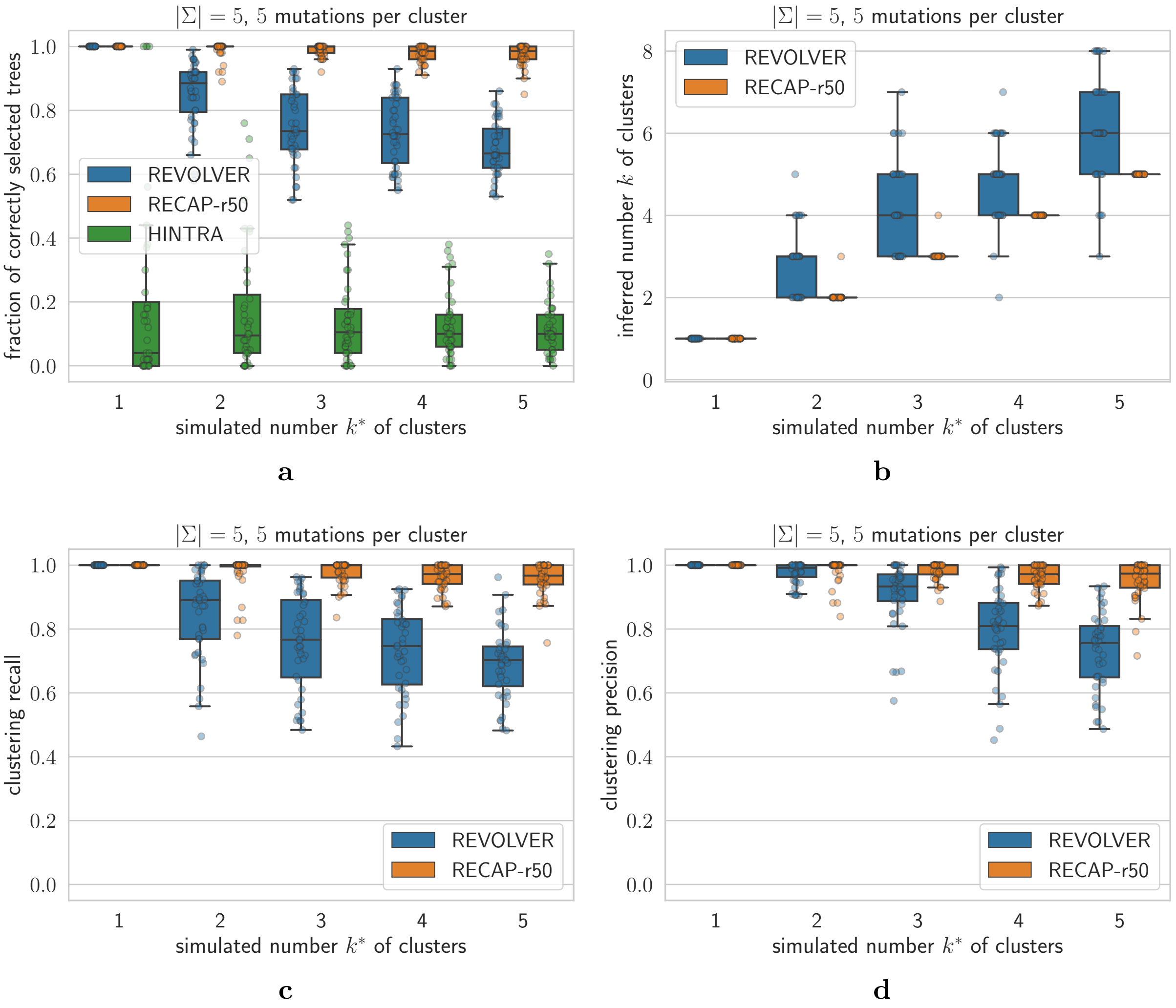


Figure 6.7: **Simulations show that RECAP accurately solves the MCCT problem for simulations with $|\Sigma| = 5$ total mutations and 5 mutations in each cluster.** (a) The fraction of patients with correctly inferred trees by each method. (b) The number k of patient clusters inferred by each method. (c) The fraction of patient pairs that are correctly clustered together. (d) The fraction of patient pairs that are correctly put in separate clusters. No results are shown in (b)-(d) for HINTRA, as this method does not infer patient clusters.

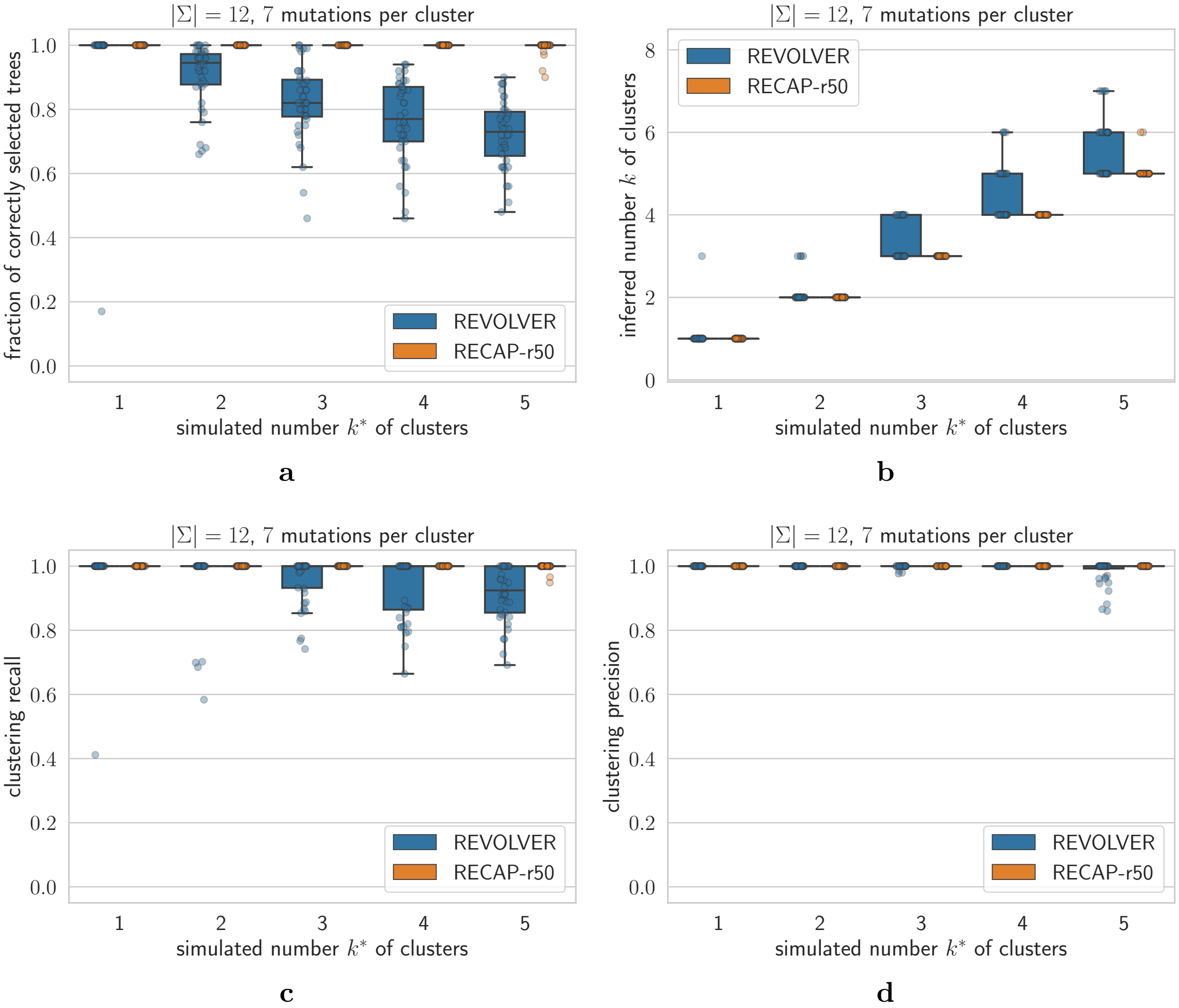


Figure 6.8: **Simulations show that RECAP accurately solves the MCCT problem for simulations with $|\Sigma| = 12$ total mutations and 7 mutations in each cluster.** (a) The fraction of patients with correctly inferred trees by each method. (b) The number k of patient clusters inferred by each method. (c) The fraction of patient pairs that are correctly clustered together. (d) The fraction of patient pairs that are correctly put in separate clusters. No results are shown in (a)-(d) for HINTRA, as this method does not infer patient clusters and does not scale to the simulated number of mutations.

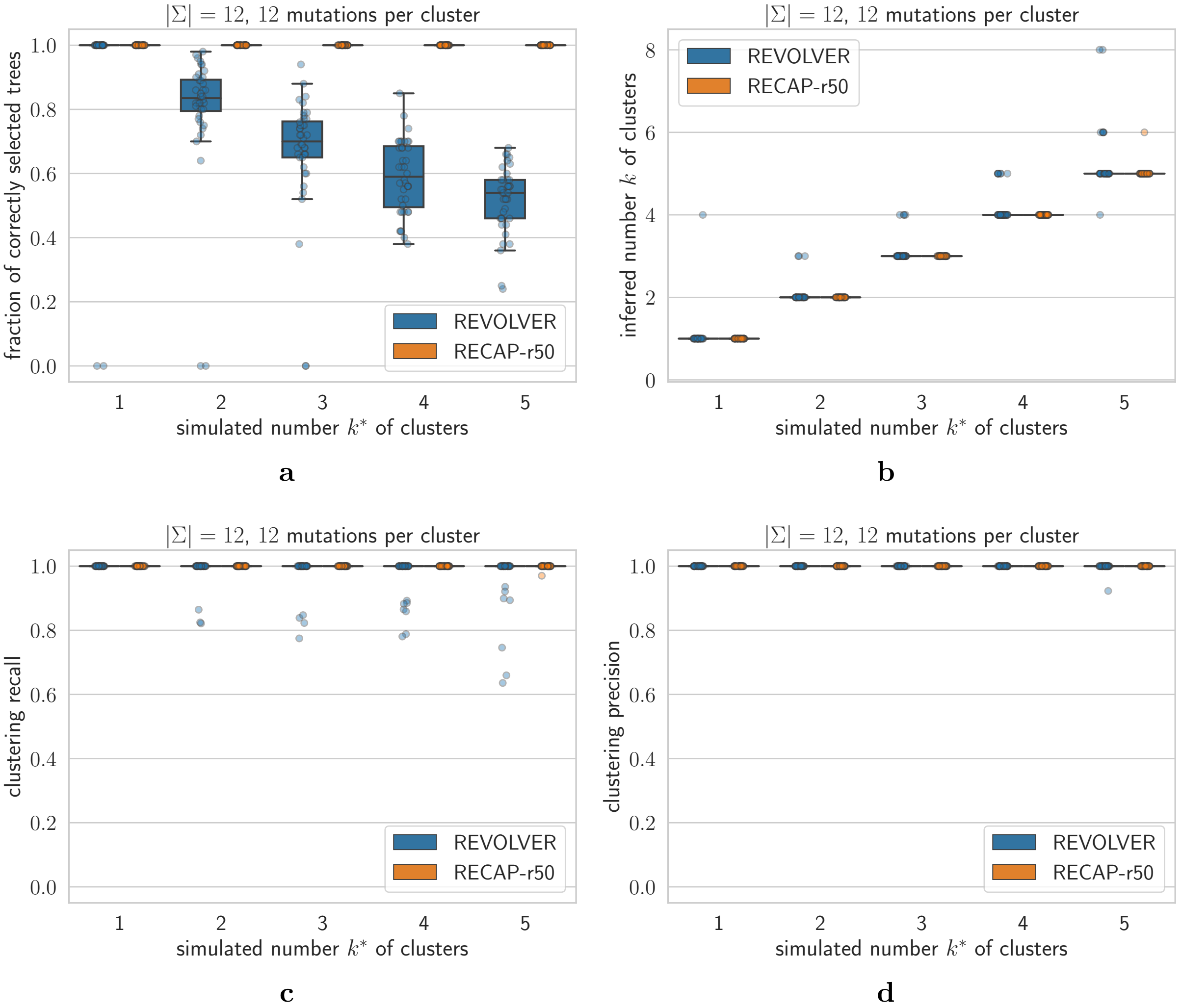


Figure 6.9: **Simulations show that RECAP accurately solves the MCCT problem for simulations with $|\Sigma| = 12$ total mutations and 12 mutations in each cluster.** (a) The fraction of patients with correctly inferred trees by each method. (b) The number k of patient clusters inferred by each method. (c) The fraction of patient pairs that are correctly clustered together. (d) The fraction of patient pairs that are correctly put in separate clusters. No results are shown in (a)-(d) for HINTRA, as this method does not infer patient clusters and does not scale to the simulated number of mutations.

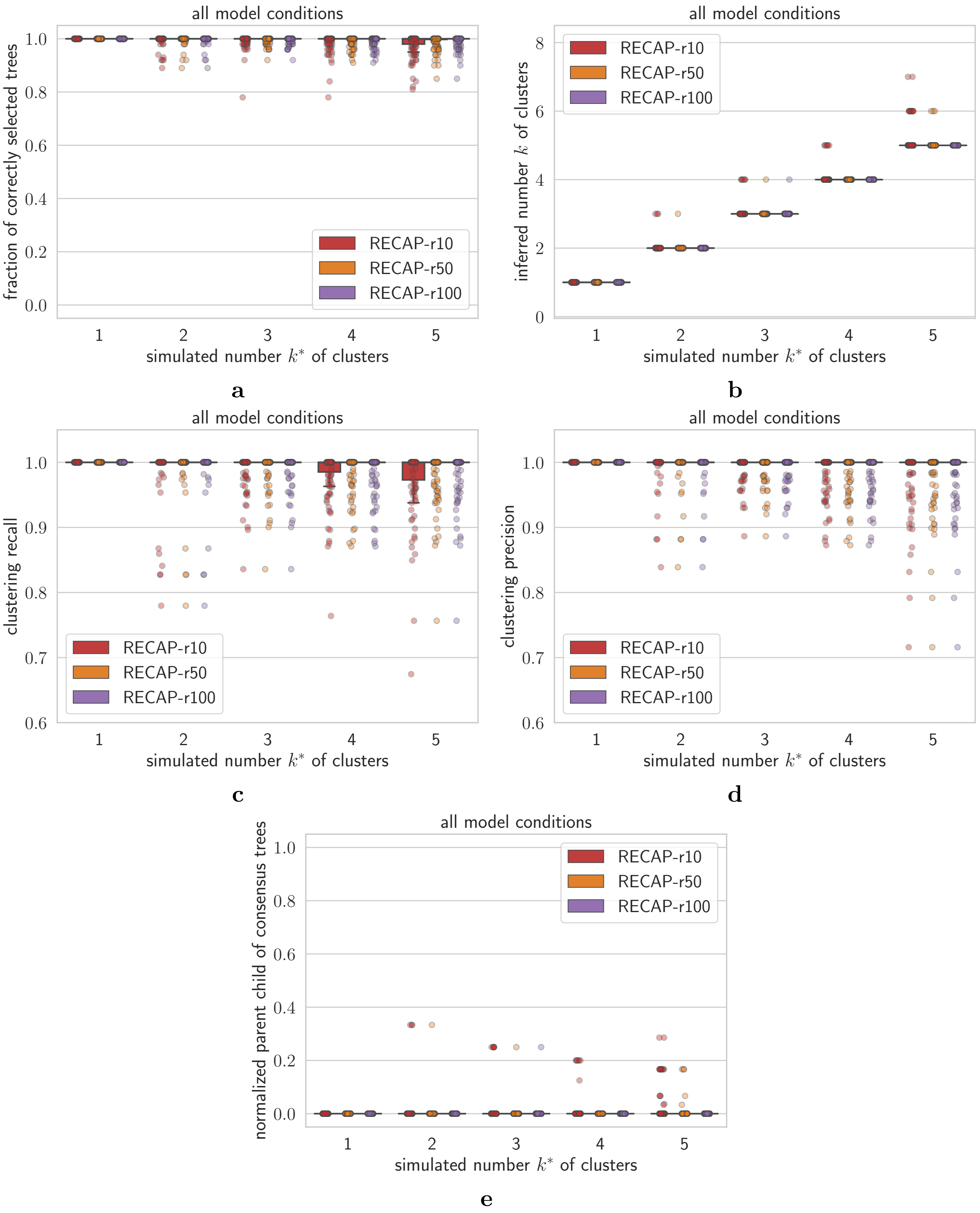


Figure 6.10: **Simulations show that performance of RECAP slightly increases with more restarts.** (a) The fraction of patients with correctly inferred trees. (b) The number k of patient clusters inferred. (c) The fraction of patient pairs that are correctly clustered together. (d) The fraction of patient pairs that are correctly put in separate clusters. (e) The normalized parent-child distances between the inferred and ground truth consensus trees.

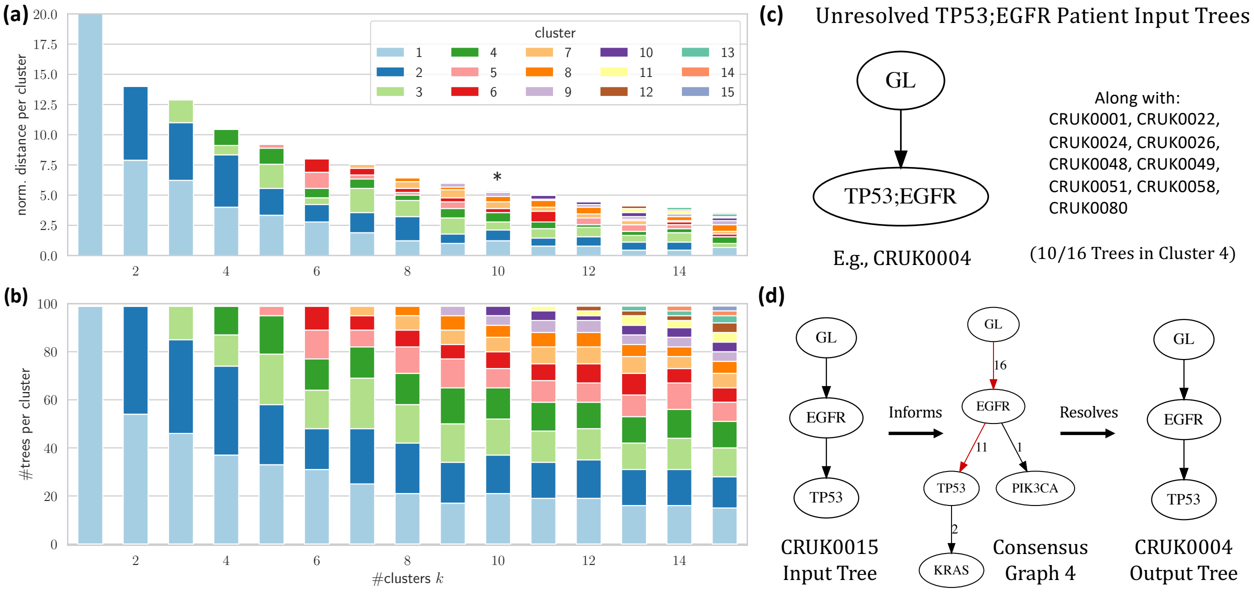
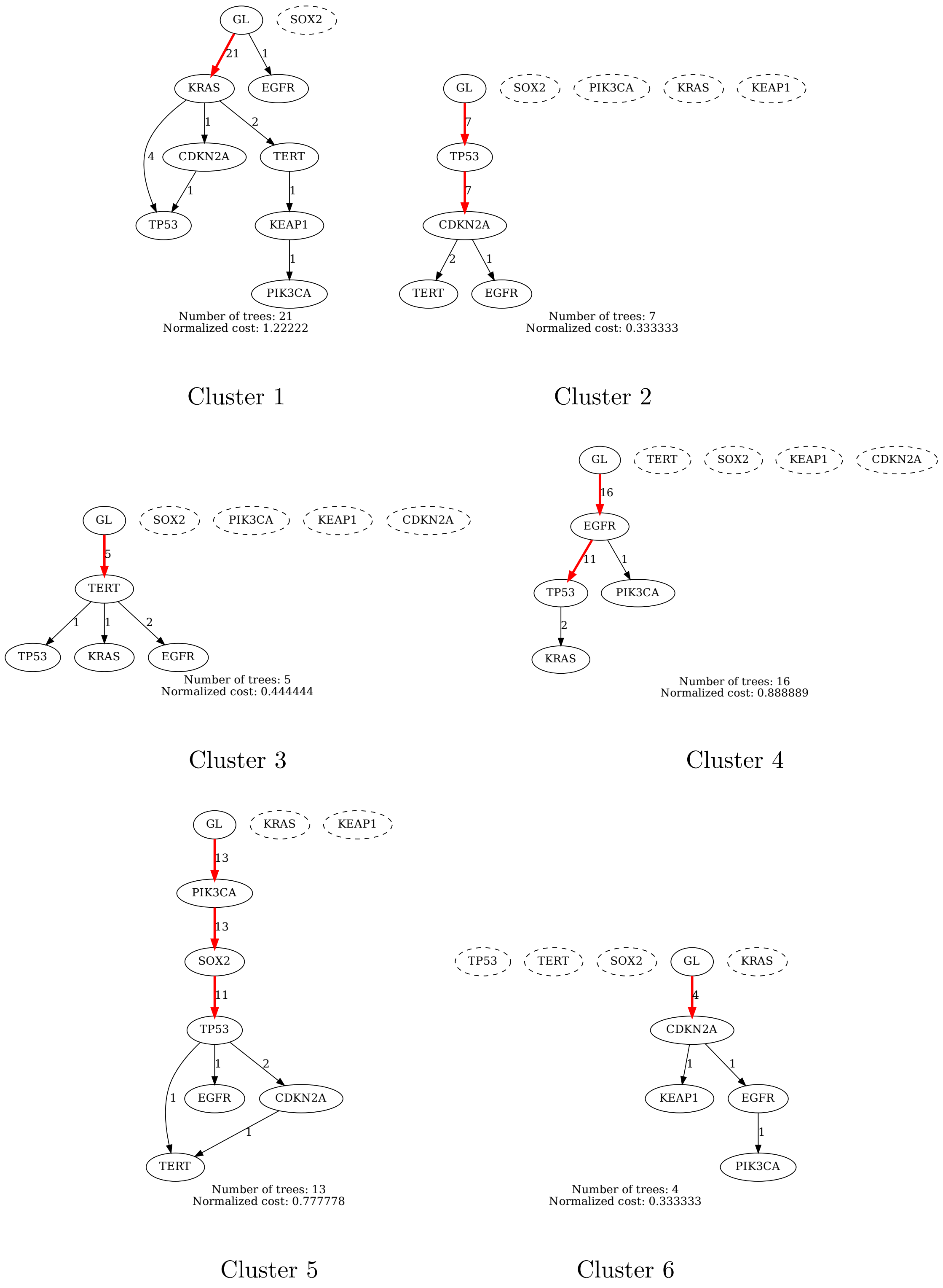


Figure 6.11: **RECAP identies repeated evolutionary patterns in a non-small cell lung cancer cohort, resolving ambiguities in the solution space and expanding mutaiton clusters.** We show results for running RECAP on TRACERx [87]. (a) The criterion scores obtained by each cluster across different values for k. As k increases, the total normalized distance decreases and levels off at $k = 10$, which RECAP selects. (b) The number of patient trees assigned to each cluster. (c) In the input data, 10 out of 16 patients that RECAP assigns to Cluster 4 have TP53 and EGFR together in a mutation cluster. (d) Patient CRUK0015 is also assigned to Cluster 4 and has an edge from EGFR to TP53. This information resolves the mutation cluster for these 10 patients via the consensus tree (red edges, edge label indicating number of patients) for this cluster.



Cluster 1

Cluster 2

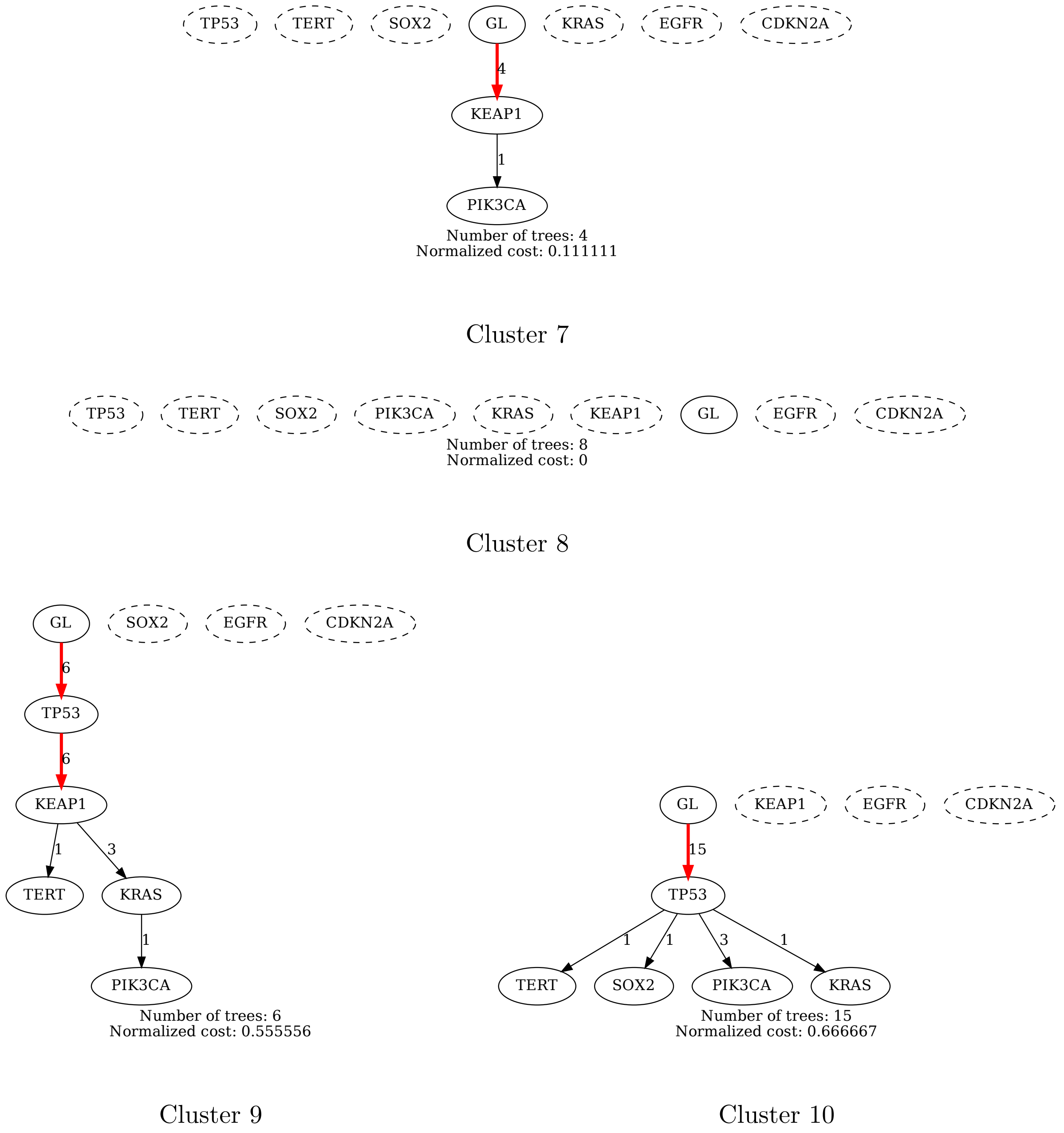
Cluster 3

Cluster 4

Cluster 5

Cluster 6

Figure 6.12: **Consensus trees identified by RECAP for a non-small cell lung cancer cohort [87].** Red edges indicate consensus tree edges, edge labels indicate number of patients with the edge. Dashed vertices indicate missing mutations. Continued in Fig. 6.13.



Cluster 7

Cluster 8

Cluster 9

Cluster 10

Figure 6.13: **Consensus trees identified by RECAP for a non-small cell lung cancer cohort [87].** Red edges indicate consensus tree edges, edge labels indicate number of patients with the edge. Dashed vertices indicate missing mutations. Note that Cluster 8 corresponds to the empty consensus tree, comprised of only the germline vertex. Continued from Fig. 6.12.

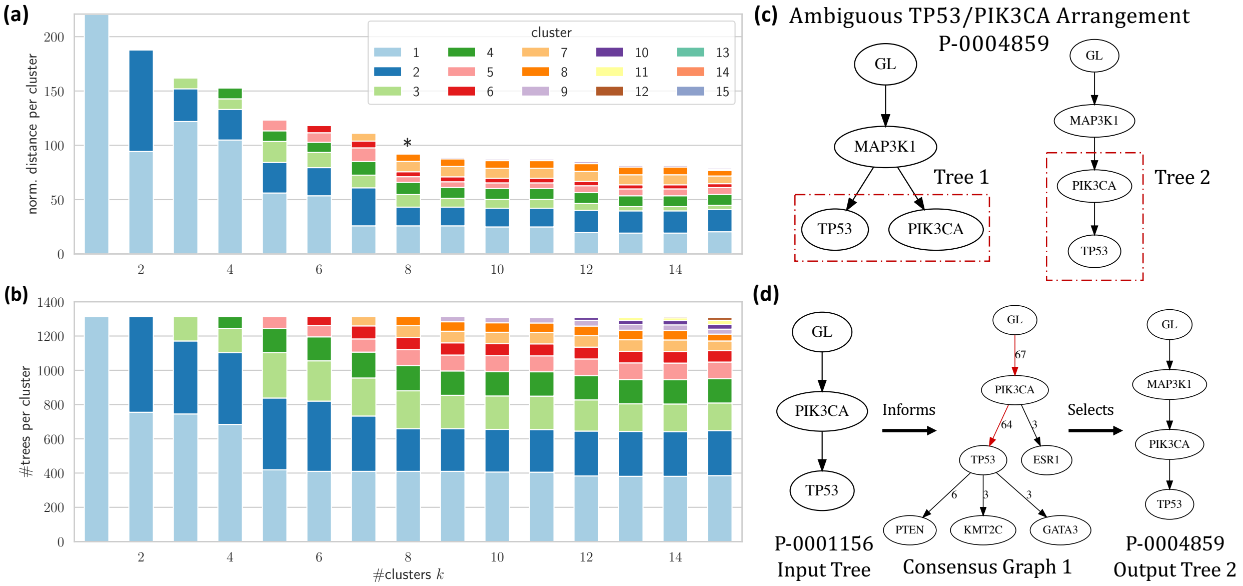
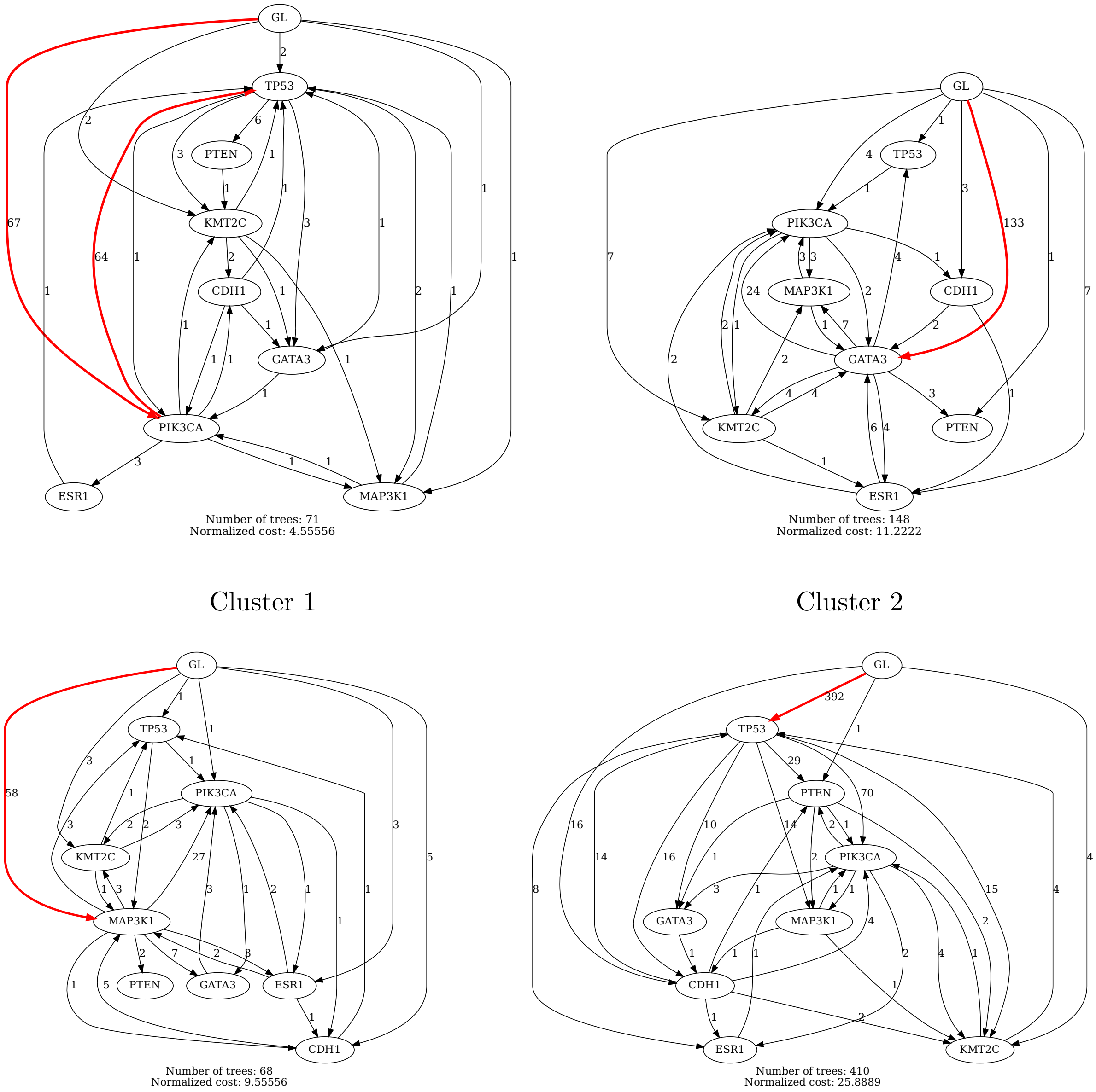


Figure 6.14: **RECAP finds a stable patient clustering and resolves ambiguities in a breast cancer cohort by identifying shared evolutionary patterns.** We show results for running RECAP on a breast cancer cohort [179]. (a) The criterion scores obtained by each cluster across different values for k. As k increases, the total normalized distance decreases and levels off at $k = 8$, which RECAP selects. (b) The number of patient trees assigned to each cluster. (c) In the input data, patient P-0004859 has two proposed trees with different arrangements of TP53 and PIK3CA. (d) This patient is assigned to Cluster 1, where other patients in this cluster have an edge from PIK3CA to TP53. Red edge coloring indicate consensus tree, and edge labels indicate the number of patients with that edge. This information is used to select the tree for P-0004859 consistent with this mutation ordering. We do not show edges in the consensus graph that occur in fewer than 3 patients in this cluster.



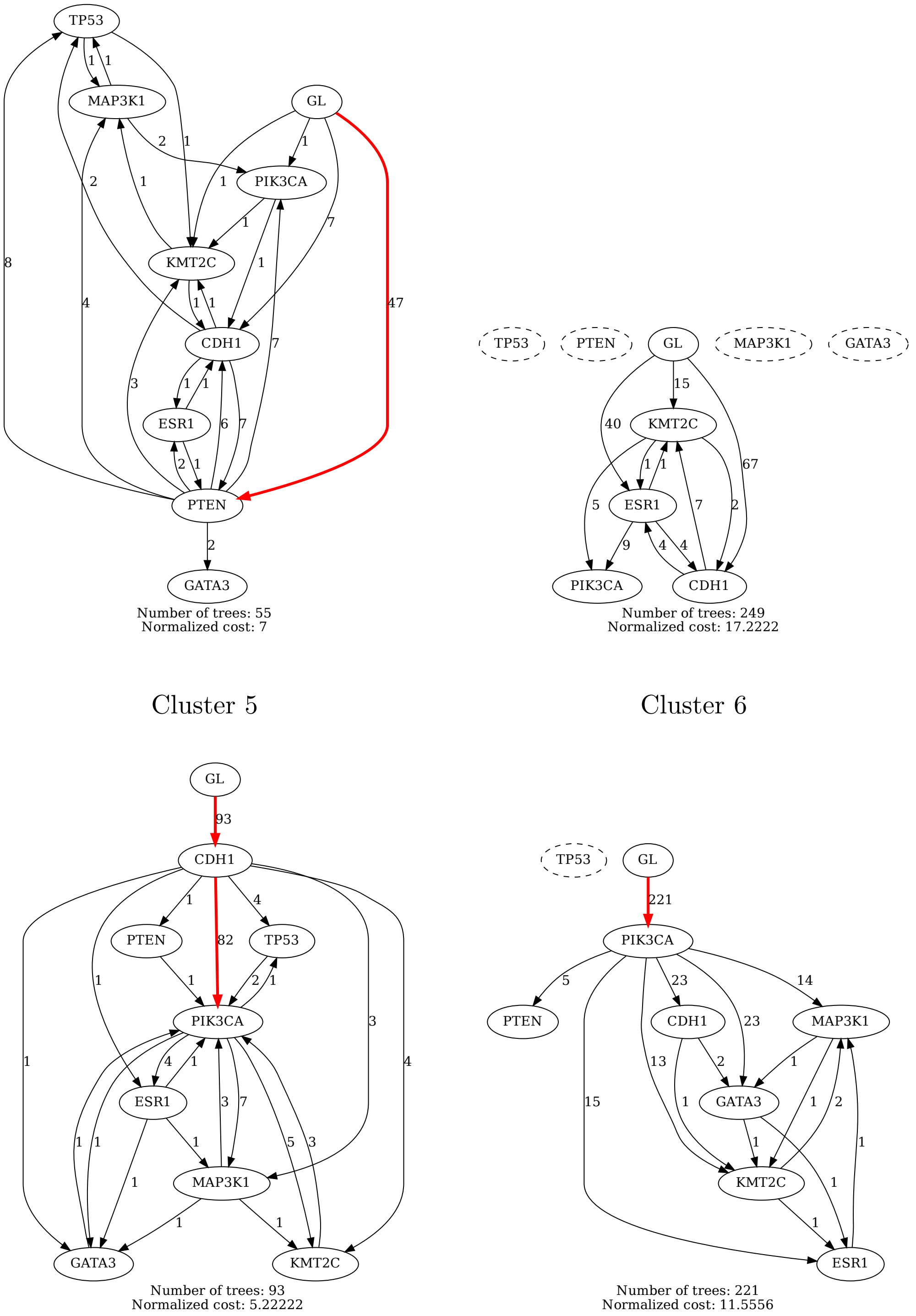
Cluster 1

Cluster 2

Cluster 3

Cluster 4

Figure 6.15: **Consensus trees identified by RECAP for a breast cancer cohort [179].** Red edges indicate consensus tree edges, edge labels indicate number of patients with the edge. Dashed vertices indicate missing mutations. Continued in Fig. 6.16.



Cluster 5

Cluster 6

Cluster 7

Cluster 8

Figure 6.16: **Consensus trees identified by RECAP for a breast cancer cohort [179].** Red edges indicate consensus tree edges, edge labels indicate number of patients with the edge. Dashed vertices indicate missing mutations. Continued from Fig. 6.15.

CHAPTER 7: CONCLUSION

In this dissertation, we introduced four methods for improving phylogeny estimation. In each case, we conceived of an optimization problem that leveraged auxiliary information to improve the quality of phylogenies estimated by current leading methods. We then established the computational complexity of this optimization problem, implemented our approach for addressing the problem, and benchmarked our empirical performance with extensive simulation studies.

In the context of species phylogenies, we introduced OCTAL and TRACTION, which jointly improve estimated gene trees by using a reference tree to add in missing species and correct weakly-supported branches. We demonstrated the utility of each method by testing it on biologically-realistic simulated datasets, showing they match or improve gene tree accuracy relative to existing methods. We conjectured that the non-parametric criterion optimized by OCTAL and TRACTION helps them to be robust to different types of gene tree heterogeneity. In the context of tumor phylogenies, we introduced PhySigs and TRACTION, which each work to reduce ambiguity in the solution space of possible evolutionary trajectories of patient tumors. In addition to prioritizing alternative evolutionary hypotheses, we were able to learn more about tumorigenesis, including the variation in exposures to mutational signatures through time and repeated evolutionary trajectories shared across patient tumors. We conjectured that integrating evolutionary context into patient subtyping will eventually help to elucidate variation in patient response to treatment and outcomes.

As we look towards future directions, we must anticipate that the influx of sequencing data will continue to grow and that the methods we develop must accommodate this data. While additional sequencing data may resolve some aspects of phylogeny reconstruction, current challenges will still persist and new challenges will emerge. For instance, despite the increase in sequencing, gene tree estimation from an alignment over just the gene of interest may continue to have weakly-supported branches, as genes are of finite length. Moreover, to capitalize on the sequencing millions of species and thousands of genes, we must be able to tolerate some amount of missing data in the MSA; otherwise, we will be stuck estimating species phylogenies over just the intersection of mutually present genes for a handful of species [49]. While OCTAL and TRACTION are early nonparametric approaches for addressing these challenges, future work should establish methods with provable guarantees under different models missing data and evolution (see [123, 124, 125]). As we expand to wider ranges of species, we will also need methods that are designed to handle multiple causes of gene tree discord (e.g., GDL, HGT, and ILS). A key component of this will likely be methods for constructing phylogenetic networks [36, 37, 38, 39]. We might think about how the optimization problems posed in OCTAL and TRACTION can be adapted to these more general graphical models under different distance metrics. Finally, we must continue to develop species phylogeny estimation methods with an eye towards downstream analysis. For example, missing data is known to impact summary methods for species tree estimation [128] so it would be helpful to measure the impact of improving gene tree estimation on species tree estimation. We should also consider the impact of improving gene tree estimation on other applications, such as adaptation inference, evolutionary event detection, ortholog identification, and analysis of functional trait evolution.

The expansion of tumor sequencing data is likewise a double edged sword. On the one hand, more tumor samples will help to reduce the solution space of possible evolutionary histories. On the other hand, current Bayesian and ILP methods will need to scale to handle larger mutation sets and more samples. Likely, methods than can efficiently integrate the strengths of bulk and single cell sequencing data will lead to cost-effective, accurate tumor phylogeny reconstruction that can be made widely accessible [116]. Complex models of evolution must also be addressed in order to accurately represent tumor development, capturing mutations other than SNVs. Crucially, methods such as PhySigs and TRACTION should be adapted to more realistic models of evolution that incorporate CNAs and loss, such as a Dollo or finite sites models [81, 82, 83]. This is especially important in certain types of cancer, such as ovarian cancer, where profuse CNAs have prevented effective genomic stratification [170]. Early work in this area is promising, but currently less computationally tractable; models that are too permissive can also be phylogenetically indecisive, making alternative optimal trees impossible to distinguish [127]. Finally, we must develop tumor phylogeny estimation methods with clinical applications in mind. To do this, we should move beyond individual tumor phylogeny inference and look for evolutionary subtypes at a patient cohort level. For instance, PhySigs may be expanded to study population-level trajectories of clonal exposures to mutational signatures. RECAP may be expanded to use other graphical structures as a consensus for each cluster, such as directed acyclic graphs incorporating mutual exclusivity of driver mutations in the same pathway. This motivates broader questions about what makes two tumor phylogenies meaningfully similar, and what distance metrics can we use to measure this similarity. It is the hope that trying these and other extensions will lead to clinically-relevant subtypes enabling precision medicine.

In conclusion, we have seen in this dissertation that evolution is a model for explaining change, whether the change observed in species or the change observed in human cells. Looking forward, there are many unknowns about how the field will progress. However, one thing we can be sure of is that, in computational biology, evolution is a constant.

REFERENCES

[1] R. Levine, “i5K: the 5,000 insect genome project,” *American Entomologist*, vol. 57, no. 2, pp. 110–113, 2011.

[2] i5K Consortium, “The i5K Initiative: Advancing arthropod genomics for knowledge, human health, agriculture, and the environment,” *Journal of Heredity*, vol. 104, no. 5, pp. 595–600, 2013.

[3] S. Cheng, M. Melkonian, S. A. Smith, S. Brockington, J. M. Archibald, P.-M. Delaux, F.-W. Li, B. Melkonian, E. V. Mavrodiev, W. Sun, Y. Fu, H. Yang, D. E. Soltis, S. W. Graham, P. S. Soltis, X. Liu, X. Xu, and G. K.-S. Wong, “10KP: A phylodiverse genome sequencing plan,” *GigaScience*, vol. 7, no. 3, 02 2018.

[4] H. A. Lewin, G. E. Robinson, W. J. Kress, W. J. Baker, J. Coddington, K. A. Crandall, R. Durbin, S. V. Edwards, F. Forest, M. T. P. Gilbert, M. M. Goldstein, I. V. Grigoriev, K. J. Hackett, D. Haussler, E. D. Jarvis, W. E. Johnson, A. Patrinos, S. Richards, J. C. Castilla-Rubio, M.-A. van Sluys, P. S. Soltis, X. Xu, H. Yang, and G. Zhang, “Earth BioGenome Project: Sequencing life for the future of life,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 17, pp. 4325–4333, 04 2018.

[5] P. C. Nowell, “The clonal evolution of tumor cell populations,” *Science*, vol. 194, pp. 23–28, 1976.

[6] R. Fisher, L. Pusztai, and C. Swanton, “Cancer heterogeneity: implications for targeted therapeutics,” *British Journal of Cancer*, vol. 108, no. 3, pp. 479–485, 2013.

[7] D. P. Tabassum and K. Polyak, “Tumorigenesis: it takes a village,” *Nature Reviews Cancer*, vol. 15, no. 8, pp. 473–483, 2015.

[8] M. Jamal-Hanjani, S. A. Quezada, J. Larkin, and C. Swanton, “Translational implications of tumor heterogeneity,” *Clinical Cancer Research*, vol. 21, no. 6, pp. 1258–1266, 2015.

[9] “The cancer genome atlas program.” [Online]. Available: https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga

[10] W. Maddison, “Gene trees in species trees,” *Systematic Biology*, vol. 46, no. 3, pp. 523–536, 1997.

[11] D. Robinson and L. Foulds, “Comparison of phylogenetic trees,” *Mathematical Biosciences*, vol. 53, no. 1-2, pp. 131–147, 1981.

[12] L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, P. J. Campbell, and M. R. Stratton, “Deciphering signatures of mutational processes operative in human cancer,” *Cell Reports*, vol. 3, no. 1, pp. 246–259, Jan. 2013.

[13] L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, S. A. J. R. Aparicio, S. Behjati, A. V. Biankin, G. R. Bignell, N. Bolli, A. Borg, A.-L. Børresen-Dale, S. Boyault, B. Burkhardt, A. P. Butler, C. Caldas, H. R. Davies, C. Desmedt, R. Eils, J. E. Eyfjörd, J. A. Foekens, M. Greaves, F. Hosoda, B. Hutter, T. Ilicic, S. Imbeaud, M. Imielinski, N. Jäger, D. T. W. Jones, D. Jones, S. Knappskog, M. Kool, S. R. Lakhani, C. López-Otín, S. Martin, N. C. Munshi, H. Nakamura, P. A. Northcott, M. Pajic, E. Papaemmanuil, A. Paradiso, J. V. Pearson, X. S. Puente, K. Raine, M. Ramakrishna, A. L. Richardson, J. Richter, P. Rosenstiel, M. Schlesner, T. N. Schumacher, P. N. Span, J. W. Teague, Y. Totoki, A. N. J. Tutt, R. Valdés-Mas, M. M. van Buuren, L. van 't Veer, A. Vincent-Salomon, N. Waddell, L. R. Yates, J. Zucman-Rossi, P. Andrew Futreal, U. McDermott, P. Lichter, M. Meyerson, S. M. Grimmond, R. Siebert, E. Campo, T. Shibata, S. M. Pfister, P. J. Campbell, M. R. Stratton, A. P. C. G. Initiative, I. B. C. Consortium, I. M.-S. Consortium, and I. Ped-Brain, "Signatures of mutational processes in human cancer," *Nature*, vol. 500, no. 7463, pp. 415–421, 2013.

[14] J. P. Huelsenbeck, J. P. Bollback, and A. M. Levine, "Inferring the root of a phylogenetic tree," *Systematic Biology*, vol. 51, no. 1, pp. 32–43, 2002.

[15] G. F. Estabrook, C. Johnson Jr, and F. R. Mc Morris, "An idealized concept of the true cladistic character," *Mathematical Biosciences*, vol. 23, no. 3-4, pp. 263–272, 1975.

[16] G. F. Estabrook, C. Johnson Jr, and F. McMorris, "A mathematical foundation for the analysis of cladistic character compatibility," *Mathematical Biosciences*, vol. 29, no. 1-2, pp. 181–187, 1976.

[17] Y. Lin, V. Rajan, and B. M. Moret, "A metric for phylogenetic trees based on matching," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 9, no. 4, pp. 1014–1022, 2012.

[18] G. F. Estabrook, F. McMorris, and C. A. Meacham, "Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units," *Systematic Zoology*, vol. 34, no. 2, pp. 193–200, 1985.

[19] E. V. Koonin, "Orthologs, paralogs, and evolutionary genomics," *Annual Review of Genetics*, vol. 39, pp. 309–338, 2005.

[20] J. C. Avise and K. Wollenberg, "Phylogenetics and the origin of species," *Proceedings of the National Academy of Sciences*, vol. 94, no. 15, pp. 7748–7755, 1997.

[21] J. B. Whitfield and P. J. Lockhart, "Deciphering ancient rapid radiations," *Trends in Ecology & Evolution*, vol. 22, no. 5, pp. 258–265, 2007.

[22] J. A. McGuire, C. W. Linkem, M. S. Koo, D. W. Hutchison, A. K. Lappin, D. I. Orange, J. Lemos-Espinal, B. R. Riddle, and J. R. Jaeger, "Mitochondrial introgression and incomplete lineage sorting through space and time: phylogenetics of crotaphytid lizards," *Evolution: International Journal of Organic Evolution*, vol. 61, no. 12, pp. 2879–2897, 2007.

[23] D. A. Pollard, V. N. Iyer, A. M. Moses, and M. B. Eisen, "Widespread discordance of gene trees with species tree in Drosophila: evidence for incomplete lineage sorting," *PLoS Genetics*, vol. 2, no. 10, p. e173, 2006.

[24] B. C. Carstens and L. L. Knowles, "Estimating species phylogeny from gene-tree probabilities despite incomplete lineage sorting: an example from Melanoplus grasshoppers," *Systematic Biology*, vol. 56, no. 3, pp. 400–411, 2007.

[25] M. V. Olson, "When less is more: gene loss as an engine of evolutionary change," *The American Journal of Human Genetics*, vol. 64, no. 1, pp. 18–23, 1999.

[26] A. Fortna, Y. Kim, E. MacLaren, K. Marshall, G. Hahn, L. Meltesen, M. Brenton, R. Hink, S. Burgers, T. Hernandez-Boussard, A. Karimpour-Fard, D. Glueck, L. McGavran, R. Berry, J. Pollack, and J. M. Sikela, "Lineage-specific gene duplication and loss in human and great ape evolution," *PLOS Biology*, vol. 2, no. 7, p. e207, 07 2004.

[27] L. H. Rieseberg, "Hybrid origins of plant species," *Annual Review of Ecology and Systematics*, vol. 28, pp. 359–389, 1997.

[28] J. Gogarten, W. Doolittle, and J. Lawrence, "Prokaryotic evolution in light of gene transfer," *Molecular Biology and Evolution*, vol. 19, no. 12, pp. 2226–2238, 2002.

[29] H. Ochman, J. G. Lawrence, and E. A. Groisman, "Lateral gene transfer and the nature of bacterial innovation," *Nature*, vol. 405, no. 6784, pp. 299–304, 2000.

[30] W. F. Doolittle, "Lateral genomics," *Trends in Biochemical Sciences*, vol. 24, no. 12, pp. M5–M8, 1999.

[31] W. F. Doolittle, "Phylogenetic classification and the universal tree," *Science*, vol. 284, no. 5423, pp. 2124–2128, 1999.

[32] C. G. Kurland, B. Canback, and O. G. Berg, "Horizontal gene transfer: a critical view," *Proceedings of the National Academy of Sciences*, vol. 100, no. 17, pp. 9658–9662, 2003.

[33] U. Bergthorsson, K. L. Adams, B. Thomason, and J. D. Palmer, "Widespread horizontal transfer of mitochondrial genes in flowering plants," *Nature*, vol. 424, no. 6945, pp. 197–201, 2003.

[34] U. Bergthorsson, A. O. Richardson, G. J. Young, L. R. Goertzen, and J. D. Palmer, "Massive horizontal transfer of mitochondrial genes from diverse land plant donors to the basal angiosperm Amborella," *Proceedings of the National Academy of Sciences*, vol. 101, no. 51, pp. 17 747–17 752, 2004.

[35] J. P. Mower, S. Stefanović, G. J. Young, and J. D. Palmer, "Gene transfer from parasitic to host plants," *Nature*, vol. 432, no. 7014, pp. 165–166, 2004.

[36] D. Huson, R. Rupp, and C. Scornavacca, *Phylogenetic networks: concepts, algorithms, and applications.* New York, NY: Cambridge University Press, 2010.

[37] D. Morrison, *Introduction to Phylogenetic Networks.* Uppsala, Sweden: RJR Productions, 2011.

[38] D. Gusfield, *ReCombinatorics: The Algorithmics of Ancestral Recombination Graphs and Explicit Phylogenetic Networks.* Cambridge, MA: MIT Press, 2014.

[39] E. Bapteste, L. van Iersel, A. Janke, S. Kelchner, S. Kelk, J. O. McInerney, D. A. Morrison, L. Nakhleh, M. Steel, L. Stougie, and J. Whitfield, "Networks: Expanding evolutionary thinking," *Trends in Genetics*, vol. 29, no. 8, pp. 439 – 441, 2013.

[40] D. Posada, "Phylogenomics for systematic biology," *Systematic Biology*, vol. 65, pp. 353–356, 2016.

[41] R. R. Hudson, "Testing the constant-rate neutral allele model with protein sequence data," *Evolution*, pp. 203–217, 1983.

[42] N. Takahata, "Gene genealogy in three related populations: consistency probability between gene and population trees." *Genetics*, vol. 122, no. 4, pp. 957–966, 1989.

[43] N. A. Rosenberg, "The probability of topological concordance of gene trees and species trees," *Theoretical Population Biology*, vol. 61, no. 2, pp. 225–247, 2002.

[44] S. Roch and M. Steel, "Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent," *Theoretical Population Biology*, vol. 100, pp. 56–62, 2015.

[45] J. Chifman and L. Kubatko, "Identifiability of the unrooted species tree topology under the coalescent model with time-reversible substitution processes, site-specific rate variation, and invariable sites," *Journal of Theoretical Biology*, vol. 374, pp. 35–47, 2015.

[46] L. Arvestad, A.-C. Berglund, J. Lagergren, and B. Sennblad, "Bayesian gene/species tree reconciliation and orthology analysis using MCMC," *Bioinformatics*, vol. 19, no. suppl_1, pp. i7–i15, 2003.

[47] M. D. Rasmussen and M. Kellis, "Unified modeling of gene duplication, loss, and coalescence using a locus tree," *Genome Research*, vol. 22, no. 4, pp. 755–765, 2012.

[48] T. Warnow, *Computational phylogenetics: an introduction to designing methods for phylogeny estimation.* Cambridge University Press, 2017.

[49] P. A. Hosner, B. C. Faircloth, T. C. Glenn, E. L. Braun, and R. T. Kimball, "Avoiding missing data biases in phylogenomic inference: An empirical study in the landfowl (Aves: Galliformes)," *Molecular Biology and Evolution*, vol. 33, no. 4, pp. 1110–1125, 2016.

[50] J. W. Streicher, J. A. Schulte, II, and J. J. Wiens, "How should genes and taxa be sampled for phylogenomic analyses with missing data? an empirical study in iguanian lizards," *Systematic Biology*, vol. 65, no. 1, p. 128, 2016.

[51] E. Noutahi, M. Semeria, M. Lafond, J. Seguin, B. Boussau, L. Guéguen, N. El-Mabrouk, and E. Tannier, “Efficient gene tree correction guided by genome evolution,” *PLoS One*, vol. 11, no. 8, p. e0159559, 2016.

[52] Y.-C. Wu, M. Rasmussen, M. Bansal, and M. Kellis, “TreeFix: Statistically informed gene tree error correction using species trees,” *Systematic Biology*, vol. 62, no. 1, pp. 110–120, 2012.

[53] M. Bansal, Y.-C. Wu, E. Alm, and M. Kellis, “Improved gene tree error correction in the presence of horizontal gene transfer,” *Bioinformatics*, vol. 31, no. 8, pp. 1211–1218, 2015.

[54] K. Chen, D. Durand, and M. Farach-Colton, “NOTUNG: A program for dating gene duplications and optimizing gene family trees,” *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 429–447, 2000.

[55] D. Durand, B. Halldórsson, and B. Vernot, “A hybrid micro-macroevolutionary approach to gene tree reconstruction,” *Journal of Computational Biology*, vol. 13, no. 2, pp. 320–335, 2006.

[56] E. Jacox, C. Chauve, G. Szöllősi, Y. Ponty, and C. Scornavacca, “ecceTERA: comprehensive gene tree-species tree reconciliation using parsimony,” *Bioinformatics*, vol. 32, no. 13, pp. 2056–2058, 2016.

[57] R. Chaudhary, J. Burleigh, and O. Eulenstein, “Efficient error correction algorithms for gene tree reconciliation based on duplication, duplication and loss, and deep coalescence,” *BMC Bioinformatics*, vol. 13, no. 10, p. S11, 2012.

[58] T. Nguyen, V. Ranwez, S. Pointet, A.-M. Chifolleau, J.-P. Doyon, and V. Berry, “Reconciliation and local gene tree rearrangement can be of mutual profit,” *Algorithms for Molecular Biology*, vol. 8, no. 1, p. 1, 2013.

[59] G. Szöllősi, W. Rosikiewicz, B. Boussau, E. Tannier, and V. Daubin, “Efficient Exploration of the Space of Reconciled Gene Trees,” *Systematic Biology*, vol. 62, no. 6, pp. 901–912, 2013.

[60] M. Lafond, C. Chauve, N. El-Mabrouk, and A. Ouangraoua, “Gene tree construction and correction using supertree and reconciliation,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 15, no. 5, pp. 1560–1570, 2018.

[61] E. Jacox, M. Weller, E. Tannier, and C. Scornavacca, “Resolution and reconciliation of non-binary gene trees with transfers, duplications and losses,” *Bioinformatics*, vol. 33, no. 7, pp. 980–987, 2017.

[62] Y. Zheng and L. Zhang, “Reconciliation with non-binary gene trees revisited,” in *Research in Computational Molecular Biology*, R. Sharan, Ed. Springer International Publishing, 2014, pp. 418–432.

[63] F.-C. Chen and W.-H. Li, "Genomic divergences between humans and other hominoids and the effective population size of the common ancestor of humans and chimpanzees," *The American Journal of Human Genetics*, vol. 68, no. 2, pp. 444–456, 2001.

[64] A. Rokas, B. L. Williams, N. King, and S. B. Carroll, "Genome-scale approaches to resolving incongruence in molecular phylogenies," *Nature*, vol. 425, no. 6960, pp. 798–804, 2003.

[65] S. R. Gadagkar, M. S. Rosenberg, and S. Kumar, "Inferring species phylogenies from multiple genes: concatenated sequence tree versus consensus gene tree," *Journal of Experimental Zoology Part B: Molecular and Developmental Evolution*, vol. 304, no. 1, pp. 64–74, 2005.

[66] M. Ruvolo, "Molecular phylogeny of the hominoids: inferences from multiple independent DNA sequence data sets," *Molecular Biology and Evolution*, vol. 14, no. 3, pp. 248–265, 1997.

[67] Y. Satta, J. Klein, and N. Takahata, "DNA archives and our nearest relative: the trichotomy problem revisited," *Molecular Phylogenetics and Evolution*, vol. 14, no. 2, pp. 259–275, 2000.

[68] E. S. Allman, J. H. Degnan, and J. A. Rhodes, "Identifying the rooted species tree from the distribution of unrooted gene trees under the coalescent," *Journal of Mathematical Biology*, vol. 62, no. 6, pp. 833–862, 2011.

[69] J. H. Degnan, "Anomalous unrooted gene trees," *Systematic Biology*, vol. 62, no. 4, pp. 574–590, 2013.

[70] S. Mirarab and T. Warnow, "ASTRAL-II: Coalescent-based species tree estimation with many hundreds of taxa and thousands of genes," *Bioinformatics*, vol. 31, no. 12, p. i44, 2015.

[71] P. Vachaspati and T. Warnow, "ASTRID: Accurate species trees from internode distances," *BMC Genomics*, vol. 16, no. 10, p. S3, 2015.

[72] O. R. Bininda-Emonds, "The evolution of supertrees," *Trends in Ecology & Evolution*, vol. 19, no. 6, pp. 315–322, 2004.

[73] J. Heled and A. Drummond, "Bayesian inference of species trees from multilocus data," *Molecular Biology and Evolution*, vol. 27, no. 3, pp. 570–580, 2010.

[74] R. Schwartz and A. A. Schäffer, "The evolution of tumour phylogenetics: principles and practice," *Nature Reviews Genetics*, vol. 18, pp. 213–229, 2017.

[75] K. Govek, C. Sikes, and L. Oesper, "A consensus approach to infer tumor evolutionary histories," in *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2018, pp. 63–72.

[76] N. Karpov, S. Malikic, M. K. Rahman, and S. C. Sahinalp, “A multi-labeled tree dissimilarity measure for comparing “clonal trees” of tumor progression,” *Algorithms for Molecular Biology*, vol. 14, no. 1, pp. 1–18, 2019.

[77] E. M. Ross and F. Markowetz, “OncoNEM: inferring tumor evolution from single-cell sequencing data,” *Genome Biology*, vol. 17, p. 69, 2016.

[78] Z. DiNardo, K. Tomlinson, A. Ritz, and L. Oesper, “Distance measures for tumor evolutionary trees,” *bioRxiv*, p. 591107, 2019.

[79] B. A. Weaver and D. W. Cleveland, “Does aneuploidy cause cancer?” *Current Opinion in Cell Biology*, vol. 18, no. 6, pp. 658–667, 2006.

[80] A. M. Taylor, J. Shih, G. Ha, G. F. Gao, X. Zhang, A. C. Berger, S. E. Schumacher, C. Wang, H. Hu, J. Liu, A. J. Lazar, S. J. Caesar-Johnson, J. A. Demchok, I. Felau, M. Kasapi, M. L. Ferguson, C. M. Hutter, H. J. Sofia, R. Tarnuzzer, Z. Wang, L. Yang, J. C. Zenklusen, J. J. Zhang, S. Chudamani, J. Liu, L. Lolla, R. Naresh, T. Pihl, Q. Sun, Y. Wan, Y. Wu, J. Cho, T. DeFreitas, S. Frazer, N. Gehlenborg, G. Getz, D. I. Heiman, J. Kim, M. S. Lawrence, ..., N. D. Aredes, A. Mariamidze, and A. D. Cherniack, “Genomic and functional approaches to understanding cancer aneuploidy,” *Cancer Cell*, vol. 33, no. 4, pp. 676–689.e3, 2018.

[81] L. Dollo, “Les lois de l’évolution,” *Bulletin de la Société Belge Géologie de Paleontolologie et d’Hydrologie*, pp. 164–166, 1893.

[82] M. El-Kebir, “SPhyR: tumor phylogeny estimation from single-cell sequencing data under loss and error,” *Bioinformatics*, vol. 34, no. 17, pp. i671–i679, 2018.

[83] G. Satas, S. Zaccaria, G. Mon, and B. J. Raphael, “SCARLET: Single-Cell Tumor Phylogeny Inference with Copy-Number Constrained Mutation Losses,” *Cell Systems*, vol. 10, no. 4, pp. 323–332, 2020.

[84] D. Hanahan and R. A. Weinberg, “The hallmarks of cancer,” *Cell*, vol. 100, no. 1, pp. 57–70, 2000.

[85] D. Hanahan and R. A. Weinberg, “Hallmarks of cancer: The next generation,” *Cell*, vol. 144, no. 5, pp. 646–674, 2011.

[86] N. McGranahan, F. Favero, E. C. de Bruin, N. J. Birkbak, Z. Szallasi, and C. Swanton, “Clonal status of actionable driver events and the timing of mutational processes in cancer evolution,” *Science Translational Medicine*, vol. 7, no. 283, pp. 283ra54–283ra54, 2015.

[87] M. Jamal-Hanjani, G. A. Wilson, N. McGranahan, N. J. Birkbak, T. B. Watkins, S. Veeriah, S. Shafi, D. H. Johnson, R. Mitter, R. Rosenthal, M. Salm, S. Horswell, M. Escudero, N. Matthews, A. Rowan, T. Chambers, D. A. Moore, S. Turajlic, H. Xu, S.-M. Lee, M. D. Forster, T. Ahmad, C. T. Hiley, C. Abbosh, M. Falzon, E. Borg, T. Marafioti, D. Lawrence, M. Hayward, S. Kolvekar, N. Panagiotopoulos, S. M. Janes, R. Thakrar, A. Ahmed, F. Blackhall, Y. Summers, R. Shah, L. Joseph, A. M. Quinn, P. A. Crosbie, B. Naidu, G. Middleton, G. Langman, S. Trotter, M. Nicolson, H. Remmen, K. Kerr, M. Chetty, L. Gomersall, D. A. Fennell, A. Nakas, S. Rathinam, G. Anand, S. Khan, P. Russell, V. Ezhil, B. Ismail, M. Irvin-Sellers, V. Prakash, J. F. Lester, M. Kornaszewska, R. Attanoos, H. Adams, H. Davies, S. Dentro, P. Taniere, B. O'Sullivan, H. L. Lowe, J. A. Hartley, N. Iles, H. Bell, Y. Ngai, J. A. Shaw, J. Herrero, Z. Szallasi, R. F. Schwarz, A. Stewart, S. A. Quezada, J. Le Quesne, P. Van Loo, C. Dive, A. Hackshaw, and C. Swanton, "Tracking the evolution of non–small-cell lung cancer," *New England Journal of Medicine*, vol. 376, no. 22, pp. 2109–2121, 2017.

[88] S. Turajlic, H. Xu, K. Litchfield, A. Rowan, S. Horswell, T. Chambers, T. O'Brien, J. I. Lopez, T. B. K. Watkins, D. Nicol, M. Stares, B. Challacombe, S. Hazell, A. Chandra, T. J. Mitchell, L. Au, C. Eichler-Jonsson, F. Jabbar, A. Soultati, S. Chowdhury, S. Rudman, J. Lynch, A. Fernando, G. Stamp, E. Nye, A. Stewart, W. Xing, J. C. Smith, M. Escudero, A. Huffman, N. Matthews, G. Elgar, B. Phillimore, M. Costa, S. Begum, S. Ward, M. Salm, S. Boeing, R. Fisher, L. Spain, C. Navas, E. Grönroos, S. Hobor, S. Sharma, I. Aurangzeb, S. Lall, A. Polson, M. Varia, C. Horsfield, N. Fotiadis, L. Pickering, R. F. Schwarz, B. Silva, J. Herrero, N. M. Luscombe, M. Jamal-Hanjani, R. Rosenthal, N. J. Birkbak, G. A. Wilson, O. Pipek, D. Ribli, M. Krzystanek, I. Csabai, Z. Szallasi, M. Gore, N. McGranahan, P. Van Loo, P. Campbell, J. Larkin, C. Swanton, and T. R. Consortium, "Deterministic evolutionary trajectories influence primary tumor growth: TRACERx renal," *Cell*, vol. 173, no. 3, pp. 595–610.e11, 04 2018.

[89] S. Turajlic, H. Xu, K. Litchfield, A. Rowan, T. Chambers, J. I. Lopez, D. Nicol, T. O'Brien, J. Larkin, S. Horswell, M. Stares, L. Au, M. Jamal-Hanjani, B. Challacombe, A. Chandra, S. Hazell, C. Eichler-Jonsson, A. Soultati, S. Chowdhury, S. Rudman, J. Lynch, A. Fernando, G. Stamp, E. Nye, F. Jabbar, L. Spain, S. Lall, R. Guarch, M. Falzon, I. Proctor, L. Pickering, M. Gore, T. B. K. Watkins, S. Ward, A. Stewart, R. DiNatale, M. F. Becerra, E. Reznik, J. J. Hsieh, T. A. Richmond, G. F. Mayhew, S. M. Hill, C. D. McNally, C. Jones, H. Rosenbaum, S. Stanislaw, D. L. Burgess, N. R. Alexander, and C. Swanton, "Tracking cancer evolution reveals constrained routes to metastases: TRACERx renal," *Cell*, vol. 173, no. 3, pp. 581–594.e12, 2018.

[90] G. Caravagna, Y. Giarratano, D. Ramazzotti, I. P. M. Tomlinson, T. A. Graham, G. Sanguinetti, and A. Sottoriva, "Detecting repeated cancer evolution from multi-region tumor sequencing data," *Nature Methods*, vol. 15, pp. 707–714, 2018.

[91] S. Khakabimamaghani, S. Malikic, J. Tang, D. Ding, R. Morin, L. Chindelevitch, and M. Ester, "Collaborative intra-tumor heterogeneity detection," *Bioinformatics*, vol. 35, no. 14, pp. i379–i388, 2019.

[92] C. A. Ortmann, D. G. Kent, J. Nangalia, Y. Silber, D. C. Wedge, J. Grinfeld, E. J. Baxter, C. E. Massie, E. Papaemmanuil, S. Menon, A. L. Godfrey, D. Dimitropoulou, P. Guglielmelli, B. Bellosillo, C. Besses, K. Döhner, C. N. Harrison, G. S. Vassiliou, A. Vannucchi, P. J. Campbell, and A. R. Green, "Effect of mutation order on myeloproliferative neoplasms," *New England Journal of Medicine*, vol. 372, no. 7, pp. 601–612, 2020/10/03 2015.

[93] H. Zafar, A. Tzen, N. Navin, K. Chen, and L. Nakhleh, "SiFit: inferring tumor trees from single-cell sequencing data under finite-sites models," *Genome Biology*, vol. 18, no. 1, pp. 1–20, 2017.

[94] P. Bonizzoni, S. Ciccolella, G. Della Vedova, and M. Soto, "Beyond perfect phylogeny: Multisample phylogeny reconstruction via ILP," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2017, pp. 1–10.

[95] D. Pradhan and M. El-Kebir, "On the non-uniqueness of solutions to the perfect phylogeny mixture problem," in *Comparative Genomics*, Oct. 2018, pp. 277–293.

[96] Y. Qi, D. Pradhan, and M. El-Kebir, "Implications of non-uniqueness in phylogenetic deconvolution of bulk DNA samples of tumors," *Algorithms for Molecular Biology*, vol. 14, no. 1, pp. 23–14, 2019.

[97] S. C. Dentro, D. C. Wedge, and P. Van Loo, "Principles of reconstructing the subclonal architecture of cancers," *Cold Spring Harbor Perspectives in Medicine*, vol. 7, no. 8, p. a026625, 2017.

[98] M. El-Kebir, L. Oesper, H. Acheson-Field, and B. J. Raphael, "Reconstruction of clonal trees and tumor composition from multi-sample sequencing data," *Bioinformatics*, vol. 31, no. 12, pp. 62–70, 2015.

[99] M. El-Kebir, G. Satas, and B. J. Raphael, "Inferring parsimonious migration histories for metastatic cancers," *Nature Genetics*, vol. 50, no. 5, pp. 718–726, May 2018.

[100] F. Strino, F. Parisi, M. Micsinai, and Y. Kluger, "TrAp: a tree approach for fingerprinting subclonal tumor composition," *Nucleic Acids Research*, vol. 41, no. 17, pp. e165–e165, 2013.

[101] M. Gerlinger, A. J. Rowan, S. Horswell, J. Larkin, D. Endesfelder, E. Gronroos, P. Martinez, N. Matthews, A. Stewart, P. Tarpey, I. Varela, B. Phillimore, S. Begum, N. Q. McDonald, A. Butler, D. Jones, K. Raine, C. Latimer, C. R. Santos, M. Nohadani, A. C. Eklund, B. Spencer-Dene, G. Clark, L. Pickering, G. Stamp, M. Gore, Z. Szallasi, J. Downward, P. A. Futreal, and C. Swanton, "Intratumor heterogeneity and branched evolution revealed by multiregion sequencing," *New England Journal of Medicine*, vol. 366, no. 10, pp. 883–892, 2012.

[102] M. Gerlinger, S. Horswell, J. Larkin, A. J. Rowan, M. P. Salm, I. Varela, R. Fisher, N. McGranahan, N. Matthews, C. R. Santos, P. Martinez, B. Phillimore, S. Begum, A. Rabinowitz, B. Spencer-Dene, S. Gulati, P. A. Bates, G. Stamp, L. Pickering, M. Gore, D. L. Nicol, S. Hazell, P. A. Futreal, A. Stewart, and C. Swanton, "Genomic architecture and evolution of clear cell renal cell carcinomas defined by multiregion sequencing," *Nature Genetics*, vol. 46, no. 3, pp. 225–233, 2014.

[103] D. E. Newburger, D. Kashef-Haghighi, Z. Weng, R. Salari, R. T. Sweeney, A. L. Brunner, S. X. Zhu, X. Guo, S. Varma, M. L. Troxell, R. B. West, S. Batzoglou, and A. Sidow, "Genome evolution during progression to breast cancer," *Genome Research*, vol. 23, no. 7, pp. 1097–1108, 07 2013.

[104] A. Schuh, J. Becq, S. Humphray, A. Alexa, A. Burns, R. Clifford, S. M. Feller, R. Grocock, S. Henderson, I. Khrebtukova, Z. Kingsbury, S. Luo, D. McBride, L. Murray, T. Menju, A. Timbs, M. Ross, J. Taylor, and D. Bentley, "Monitoring chronic lymphocytic leukemia progression by whole genome sequencing reveals heterogeneous clonal evolution patterns," *Blood, The Journal of the American Society of Hematology*, vol. 120, no. 20, pp. 4191–4196, 11 2012.

[105] A. G. Deshwar, S. Vembu, C. K. Yung, G. H. Jang, L. Stein, and Q. Morris, "PhyloWGS: reconstructing subclonal composition and evolution from whole-genome sequencing of tumors," *Genome Biology*, vol. 16, no. 1, pp. 1–20, 2015.

[106] W. Jiao, S. Vembu, A. G. Deshwar, L. Stein, and Q. Morris, "Inferring clonal evolution of tumors from single nucleotide somatic mutations," *BMC Bioinformatics*, vol. 15, no. 1, p. 35, 2014.

[107] S. Malikic, A. W. McPherson, N. Donmez, and C. S. Sahinalp, "Clonality inference in multiple tumor samples using phylogeny," *Bioinformatics*, vol. 31, no. 9, pp. 1349–1356, 2015.

[108] V. Popic, R. Salari, I. Hajirasouliha, D. Kashef-Haghighi, R. B. West, and S. Batzoglou, "Fast and scalable inference of multi-sample cancer lineages," *Genome Biology*, vol. 16, no. 1, p. 91, 2015.

[109] M. El-Kebir, G. Satas, L. Oesper, and B. J. Raphael, "Inferring the mutational history of a tumor using multi-state perfect phylogeny mixtures," *Cell Systems*, vol. 3, no. 1, pp. 43–53, 2016.

[110] N. E. Navin, "Cancer genomics: one cell at a time," *Genome Biology*, vol. 15, no. 8, p. 452, 2014.

[111] Y. Hou, L. Song, P. Zhu, B. Zhang, Y. Tao, X. Xu, F. Li, K. Wu, J. Liang, D. Shao, H. Wu, X. Ye, C. Ye, R. Wu, M. Jian, Y. Chen, W. Xie, R. Zhang, L. Chen, X. Liu, X. Yao, H. Zheng, C. Yu, Q. Li, Z. Gong, M. Mao, X. Yang, L. Yang, J. Li, W. Wang, Z. Lu, N. Gu, G. Laurie, L. Bolund, K. Kristiansen, J. Wang, H. Yang, Y. Li, X. Zhang, and J. Wang, "Single-cell exome sequencing and monoclonal evolution of a JAK2-negative myeloproliferative neoplasm," *Cell*, vol. 148, no. 5, pp. 873–885, 2012.

[112] X. Xu, Y. Hou, X. Yin, L. Bao, A. Tang, L. Song, F. Li, S. Tsang, K. Wu, H. Wu, W. He, L. Zeng, M. Xing, R. Wu, H. Jiang, X. Liu, D. Cao, G. Guo, X. Hu, Y. Gui, Z. Li, W. Xie, X. Sun, M. Shi, Z. Cai, B. Wang, M. Zhong, J. Li, Z. Lu, N. Gu, X. Zhang, L. Goodman, L. Bolund, J. Wang, H. Yang, K. Kristiansen, M. Dean, Y. Li, and J. Wang, "Single-cell exome sequencing reveals single-nucleotide mutation characteristics of a kidney tumor," *Cell*, vol. 148, no. 5, pp. 886–895, 2012.

[113] K. I. Kim and R. Simon, "Using single cell sequencing data to model the evolutionary history of a tumor," *BMC Bioinformatics*, vol. 15, no. 1, p. 27, 2014.

[114] K. Jahn, J. Kuipers, and N. Beerenwinkel, "Tree inference for single-cell data," *Genome Biology*, vol. 17, no. 1, pp. 1–17, 2016.

[115] S. Malikic, K. Jahn, J. Kuipers, S. C. Sahinalp, and N. Beerenwinkel, "Integrative inference of subclonal tumour evolution from single-cell and bulk sequencing data," *Nature Communications*, vol. 10, no. 1, pp. 1–12, 2019.

[116] L. Weber, N. Aguse, N. Chia, and M. El-Kebir, "PhyDOSE: Design of follow-up single-cell sequencing experiments of tumors," in *Computational Cancer Biology: International Workshop, RECOMB CCB 2020, Virtual, June 18-19, 2020, Proceedings*. Springer International Publishing, 2020.

[117] S. Christensen, E. K. Molloy, P. Vachaspati, and T. Warnow, "Optimal completion of incomplete gene trees in polynomial time using OCTAL," in *17th International Workshop on Algorithms in Bioinformatics (WABI 2017)*, R. Schwartz and K. Reinert, Eds., vol. 88, 2017, pp. 27:1–27:14.

[118] S. Christensen, E. K. Molloy, P. Vachaspati, and T. Warnow, "OCTAL: Optimal completion of gene trees in polynomial time," *Algorithms for Molecular Biology*, vol. 13, no. 1, p. 6, 2018.

[119] Z. Xi, L. Liu, and C. C. Davis, "The impact of missing data on species tree estimation," *Molecular Biology and Evolution*, vol. 33, no. 3, pp. 838–860, 2016.

[120] M. Kennedy and R. D. Page, "Seabird supertrees: Combining partial estimates of Procellariiform phylogeny," *The Auk*, vol. 119, no. 1, pp. 88–108, 2002.

[121] J. G. Burleigh, K. W. Hilu, and D. E. Soltis, "Inferring phylogenies with incomplete data sets: A 5-gene, 567-taxon analysis of angiosperms," *BMC Evolutionary Biology*, vol. 9, no. 1, p. 61, 2009.

[122] E. S. Allman, J. H. Degnan, and J. A. Rhodes, "Split probabilities and species tree inference under the multispecies coalescent model," *arXiv:1704.04268*, 2017.

[123] M. Nute and J. Chou, "Statistical consistency of coalescent-based species tree methods under models of missing data," *Comparative Genomics: 15th International Workshop, RECOMB CG 2017, Barcelona, Spain, October 4-6, 2017, Proceedings*, pp. 277–297, 2017.

[124] M. Nute, J. Chou, E. K. Molloy, and T. Warnow, "The performance of coalescent-based species tree estimation methods under models of missing data," *BMC Genomics*, vol. 19, no. 5, pp. 1–22, 2018.

[125] J. A. Rhodes, M. G. Nute, and T. Warnow, "NJst and ASTRID are not statistically consistent under a random model of missing data," *arXiv preprint arXiv:2001.07844*, 2020.

[126] Huang, Huateng and Knowles, L. Lacey, "Unforeseen consequences of excluding missing data from next-generation sequences: Simulation study of RAD sequences," *Systematic Biology*, vol. 65, no. 3, pp. 357–365, 2016.

[127] M. J. Sanderson, M. M. McMahon, and M. Steel, "Phylogenomics with incomplete taxon coverage: the limits to inference," *BMC Evolutionary Biology*, vol. 10, 2010.

[128] E. Molloy and T. Warnow, "To include or not to include: The impact of gene filtering on species tree estimation methods," *Systematic Biology*, vol. 67, no. 2, pp. 285–303, 2018.

[129] S. Mir arabbaygi (Mirarab), "Novel scalable approaches for multiple sequence alignment and phylogenomic reconstruction," Ph.D. dissertation, The University of Texas at Austin, 2015. [Online]. Available: http://hdl.handle.net/2152/31377

[130] D. Mallo, L. Martins, and D. Posada, "SimPhy: phylogenomic simulation of gene, locus, and species trees," *Systematic Biology*, vol. 65, no. 2, pp. 334–344, 2016.

[131] W. Fletcher and Z. Yang, "INDELible: A flexible simulator of biological sequence evolution," *Molecular Biology and Evolution*, vol. 26, no. 8, pp. 1879–1888, 2009, 10.1093/molbev/msp098.

[132] J. Sukumaran and M. Holder, "Dendropy: a Python library for phylogenetic computing," *Bioinformatics*, vol. 26, no. 12, pp. 1569–1571, 2010.

[133] A. Stamatakis, "RAxML Version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies," *Bioinformatics*, vol. 30, no. 9, 2014.

[134] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[135] O. J. Dunn, "Multiple comparisons among means," *Journal of the American Statistical Association*, vol. 56, no. 293, pp. 52–64, 1961.

[136] J. Sukumaran and M. Holder, "The DendroPy phylogenetic computing library documentation: Trees," http://dendropy.readthedocs.io/en/latest/library/treemodel.html, accessed: 2017-10-20.

[137] M. Suchard and B. Redelings, "BAli-Phy: simultaneous Bayesian inference of alignment and phylogeny," *Bioinformatics*, vol. 22, pp. 2047–2048, 2006.

[138] T. Mailund and C. N. Pedersen, "QDist-quartet distance between evolutionary trees," *Bioinformatics*, vol. 20, no. 10, pp. 1636–1637, 2004.

[139] Y. Lin, V. Rajan, and B. Moret, "Software for the matching distance of Lin, Rajan, and Moret," 2018, available at http://users.cecs.anu.edu.au/~u1024708/index_files/matching_distance.zip.

[140] C. Zhang, E. Sayyari, and S. Mirarab, *ASTRAL-III: Increased Scalability and Impacts of Contracting Low Support Branches.* Springer International Publishing, 2017, pp. 53–75.

[141] M. S. Bayzid and T. Warnow, "Gene tree parsimony for incomplete gene trees," in *17th International Workshop on Algorithms in Bioinformatics (WABI 2017)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), R. Schwartz and K. Reinert, Eds., vol. 88. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, pp. 2:1–2:13.

[142] M. K. Kuhner and J. Felsenstein, "A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates," *Molecular Biology and Evolution*, vol. 11, no. 3, pp. 459–468, 1994.

[143] L. J. Billera, S. P. Holmes, and K. Vogtmann, "Geometry of the space of phylogenetic trees," *Advances in Applied Mathematics*, vol. 27, no. 4, pp. 733 – 767, 2001.

[144] S. Christensen, E. K. Molloy, P. Vachaspati, and T. Warnow, "TRACTION: Fast non-parametric improvement of estimated gene trees," in *19th International Workshop on Algorithms in Bioinformatics (WABI 2019)*, K. T. Huber and D. Gusfield, Eds., vol. 143, 2019, pp. 4:1–4:16.

[145] S. Christensen, E. K. Molloy, P. Vachaspati, A. Yammanuru, and T. Warnow, "Non-parametric correction of estimated gene trees using TRACTION," *Algorithms for Molecular Biology*, vol. 15, no. 1, pp. 1–18, 2020.

[146] E. D. Jarvis, S. Mirarab, A. J. Aberer, B. Li, P. Houde, C. Li, S. Y. W. Ho, B. C. Faircloth, B. Nabholz, J. T. Howard, A. Suh, C. C. Weber, R. R. da Fonseca, J. Li, F. Zhang, H. Li, L. Zhou, N. Narula, L. Liu, G. Ganapathy, B. Boussau, M. S. Bayzid, V. Zavidovych, S. Subramanian, T. Gabaldón, S. Capella-Gutiérrez, J. Huerta-Cepas, B. Rekepalli, K. Munch, M. Schierup, B. Lindow, W. C. Warren, D. Ray, R. E. Green, M. W. Bruford, X. Zhan, ..., T. Warnow, W. Jun, M. T. P. Gilbert, and G. Zhang, "Whole-genome analyses resolve early branches in the tree of life of modern birds," *Science*, vol. 346, no. 6215, pp. 1320–1331, 12 2014.

[147] B. Vernot, M. Stolzer, A. Goldman, and D. Durand, "Reconciliation with non-binary species trees," *Journal of Computational Biology*, vol. 15, no. 8, pp. 981–1006, 2008.

[148] R. Chaudhary, J. G. Burleigh, and D. Fernández-Baca, "Inferring species trees from incongruent multi-copy gene trees using the Robinson-Foulds distance," *Algorithms for Molecular Biology*, vol. 8, no. 1, p. 28, 2013.

[149] M. S. Bansal, "Linear-time algorithms for some phylogenetic tree completion problems under Robinson-Foulds distance," in *Comparative Genomics*, M. Blanchette and A. Ouangraoua, Eds. Springer International Publishing, 2018, pp. 209–226.

[150] P. Gawrychowski, G. Landau, W.-K. Sung, and O. Weimann, "A faster construction of phylogenetic consensus trees," *arXiv preprint arXiv:1705.10548*, 2017.

[151] G. Ganapathy, B. Goodson, R. Jansen, H.-s. Le, V. Ramachandran, and T. Warnow, "Pattern identification in biogeography," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 3, no. 4, pp. 334–346, 2006.

[152] R. Davidson, P. Vachaspati, S. Mirarab, and T. Warnow, "Phylogenomic species tree estimation in the presence of incomplete lineage sorting and horizontal gene transfer," *BMC Genomics*, vol. 16, p. S1, 2015.

[153] V. Lefort, R. Desper, and O. Gascuel, "FastME 2.0: A comprehensive, accurate, and fast distance-based phylogeny inference program," *Molecular Biology and Evolution*, vol. 32, no. 10, pp. 2798–2800, 2015.

[154] M. Lafond, M. Semeria, K. M. Swenson, E. Tannier, and N. El-Mabrouk, "Gene tree correction guided by orthology," *BMC Bioinformatics*, vol. 14, no. 15, p. S5, Oct 2013.

[155] H. Lai, M. Stolzer, and D. Durand, "Fast heuristics for resolving weakly supported branches using duplication, transfers, and losses," in *Comparative Genomics*, J. Meidanis and L. Nakhleh, Eds. Springer International Publishing, 2017, pp. 298–320.

[156] S. Christensen, M. D. Leiserson, and M. El-Kebir, "PhySigs: Phylogenetic inference of mutational signature dynamics," in *Pacific Symposium on Biocomputing (PSB)*, vol. 25, 2020, pp. 226–237.

[157] A. V. Hoeck, N. H. Tjoonk, R. v. Boxtel, and E. Cuppen, "Portrait of a cancer: mutational signature analyses for cancer diagnostics," *BMC Cancer*, vol. 19, p. 457, 2019.

[158] H. Davies, D. Glodzik, S. Morganella, L. R. Yates, J. Staaf, X. Zou, M. Ramakrishna, S. Martin, S. Boyault, A. M. Sieuwerts, P. T. Simpson, T. A. King, K. Raine, J. E. Eyfjord, G. Kong, Å. Borg, E. Birney, H. G. Stunnenberg, M. J. van de Vijver, A.-L. Børresen-Dale, J. W. M. Martens, P. N. Span, S. R. Lakhani, A. Vincent-Salomon, C. Sotiriou, A. Tutt, A. M. Thompson, S. Van Laere, A. L. Richardson, A. Viari, P. J. Campbell, M. R. Stratton, and S. Nik-Zainal, "HRDetect is a predictor of BRCA1 and BRCA2 deficiency based on mutational signatures," *Nature Medicine*, vol. 23, no. 4, pp. 517–525, 04 2017.

[159] R. Rosenthal, N. McGranahan, J. Herrero, B. S. Taylor, and C. Swanton, "deconstructSigs: delineating mutational processes in single tumors distinguishes DNA repair deficiencies and patterns of carcinoma evolution," *Genome Biology*, vol. 17, no. 1, p. 31, Dec. 2016.

[160] X. Huang, D. Wojtowicz, and T. M. Przytycka, “Detecting presence of mutational signatures in cancer with confidence,” *Bioinformatics*, vol. 34, no. 2, pp. 330–337, Sep. 2017.

[161] F. Blokzijl, R. Janssen, R. van Boxtel, and E. Cuppen, “MutationalPatterns: comprehensive genome-wide analysis of mutational processes,” *Genome Medicine*, vol. 10, no. 1, pp. 1–11, Dec. 2018.

[162] A. McPherson, A. Roth, E. Laks, T. Masud, A. Bashashati, A. W. Zhang, G. Ha, J. Biele, D. Yap, A. Wan, L. M. Prentice, J. Khattra, M. A. Smith, C. B. Nielsen, S. C. Mullaly, S. Kalloger, A. Karnezis, K. Shumansky, C. Siu, J. Rosner, H. L. Chan, J. Ho, N. Melnyk, J. Senz, W. Yang, R. Moore, A. J. Mungall, M. A. Marra, A. Bouchard-Côté, C. B. Gilks, D. G. Huntsman, J. N. McAlpine, S. Aparicio, and S. P. Shah, “Divergent modes of clonal spread and intraperitoneal mixing in high-grade serous ovarian cancer,” *Nature Genetics*, vol. 48, no. 7, pp. 758–767, 2016.

[163] Y. Rubanova, R. Shi, R. Li, J. Wintersinger, N. Sahin, A. Deshwar, Q. Morris, P. Evolution, H. W. Group, and P. network, “TrackSig: reconstructing evolutionary trajectories of mutations in cancer,” *bioRxiv*, p. 260471, nov 2018.

[164] M. Petljak, L. B. Alexandrov, J. S. Brammeld, S. Price, D. C. Wedge, S. Grossmann, K. J. Dawson, Y. S. Ju, F. Iorio, J. M. C. Tubio, C. C. Koh, I. Georgakopoulos-Soares, B. Rodríguez-Martín, B. Otlu, S. O’Meara, A. P. Butler, A. Menzies, S. G. Bhosle, K. Raine, D. R. Jones, J. W. Teague, K. Beal, C. Latimer, L. O’Neill, J. Zamora, E. Anderson, N. Patel, M. Maddison, B. L. Ng, J. Graham, M. J. Garnett, U. McDermott, S. Nik-Zainal, P. J. Campbell, and M. R. Stratton, “Characterizing mutational signatures in human cancer cell lines reveals episodic APOBEC mutagenesis,” *Cell*, vol. 176, no. 6, pp. 1282–1294.e20, 2019.

[165] J. G. Tate, S. Bamford, H. C. Jubb, Z. Sondka, D. M. Beare, N. Bindal, H. Boutselakis, C. G. Cole, C. Creatore, E. Dawson, P. Fish, B. Harsha, C. Hathaway, S. C. Jupe, C. Y. Kok, K. Noble, L. Ponting, C. C. Ramshaw, C. E. Rye, H. E. Speedy, R. Stefancsik, S. L. Thompson, S. Wang, S. Ward, P. J. Campbell, and S. A. Forbes, “COSMIC: the catalogue of somatic mutations in cancer,” *Nucleic Acids Research*, vol. 47, no. D1, pp. D941–D947, 10 2018.

[166] J. Kim, K. W. Mouw, P. Polak, L. Z. Braunstein, A. Kamburov, G. Tiao, D. J. Kwiatkowski, J. E. Rosenberg, E. M. Van Allen, A. D. D’Andrea, and G. Getz, “Somatic ERCC2 mutations are associated with a distinct genomic signature in urothelial tumors,” *Nature Genetics*, vol. 48, no. 6, pp. 600–606, 2016.

[167] L. D. Trucco, P. A. Mundra, K. Hogan, P. Garcia-Martinez, A. Viros, A. K. Mandal, N. Macagno, C. Gaudy-Marqueste, D. Allan, F. Baenke, M. Cook, C. McManus, B. Sanchez-Laorden, N. Dhomen, and R. Marais, “Ultraviolet radiation–induced DNA damage is prognostic for outcome in melanoma,” *Nature Medicine*, vol. 25, no. 2, pp. 221–224, 2019.

[168] G. Schwarz, “Estimating the dimension of a model,” *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, Mar. 1978.

[169] “Signatures v2 matrix.” [Online]. Available: ”https://cancer.sanger.ac.uk/signatures_v2/matrix.png”

[170] G. Macintyre, T. E. Goranova, D. De Silva, D. Ennis, A. M. Piskorz, M. Eldridge, D. Sie, L.-A. Lewsley, A. Hanif, C. Wilson, S. Dowson, R. M. Glasspool, M. Lockley, E. Brockbank, A. Montes, A. Walther, S. Sundar, R. Edmondson, G. D. Hall, A. Clamp, C. Gourley, M. Hall, C. Fotopoulou, H. Gabra, J. Paul, A. Supernat, D. Millan, A. Hoyle, G. Bryson, C. Nourse, L. Mincarelli, L. N. Sanchez, B. Ylstra, M. Jimenez-Linan, L. Moore, O. Hofmann, F. Markowetz, I. A. McNeish, and J. D. Brenton, “Copy number signatures and mutational processes in ovarian carcinoma,” *Nature Genetics*, vol. 50, no. 9, pp. 1262–1270, 2018.

[171] L. B. Alexandrov, P. H. Jones, D. C. Wedge, J. E. Sale, P. J. Campbell, S. Nik-Zainal, and M. R. Stratton, “Clock-like mutational processes in human somatic cells,” *Nature Genetics*, vol. 47, no. 12, pp. 1402–1407, 12 2015.

[172] J. T. den Dunnen, R. Dalgleish, D. R. Maglott, R. K. Hart, M. S. Greenblatt, J. McGowan-Jordan, A.-F. Roux, T. Smith, S. E. Antonarakis, and P. E. M. Taschner, “HGVS recommendations for the description of sequence variants: 2016 update,” *Human Mutation*, vol. 37, no. 6, pp. 564–569, 2016.

[173] W. McLaren, L. Gil, S. E. Hunt, H. S. Riat, G. R. Ritchie, A. Thormann, P. Flicek, and F. Cunningham, “The ensembl variant effect predictor,” *Genome Biology*, vol. 17, no. 1, p. 122, 2016.

[174] D. Chakravarty, J. Gao, S. Phillips, R. Kundra, H. Zhang, J. Wang, J. E. Rudolph, R. Yaeger, T. Soumerai, M. H. Nissan, M. T. Chang, S. Chandarlapaty, T. A. Traina, P. K. Paik, A. L. Ho, F. M. Hantash, A. Grupe, S. S. Baxi, M. K. Callahan, A. Snyder, P. Chi, D. C. Danila, M. Gounder, J. J. Harding, M. D. Hellmann, G. Iyer, Y. Y. Janjigian, T. Kaley, D. A. Levine, M. Lowery, A. Omuro, M. A. Postow, D. Rathkopf, A. N. Shoushtari, N. Shukla, M. H. Voss, E. Paraiso, A. Zehir, M. F. Berger, B. S. Taylor, L. B. Saltz, G. J. Riely, M. Ladanyi, D. M. Hyman, J. Baselga, P. Sabbatini, D. B. Solit, and N. Schultz, “OncoKB: A precision oncology knowledge base,” *JCO Precision Oncology*, no. 1, pp. 1–16, 2020/10/03 2017.

[175] S. Christensen, J. Kim, N. Chia, O. Koyejo, and M. El-Kebir, “Detecting evolutionary patterns of cancers using consensus trees,” in *European Conference on Computational Biology (ECCB)*, 2020.

[176] C. Curtis, S. P. Shah, S.-F. Chin, G. Turashvili, O. M. Rueda, M. J. Dunning, D. Speed, A. G. Lynch, S. Samarajiwa, Y. Yuan, S. Graf, G. Ha, G. Haffari, A. Bashashati, R. Russell, S. McKinney, M. Group, A. Langerød, A. Green, E. Provenzano, G. Wishart, S. Pinder, P. Watson, F. Markowetz, L. Murphy, I. Ellis, A. Purushotham, A.-L. Børresen-Dale, J. D. Brenton, S. Tavaré, C. Caldas, , and S. Aparicio, "Dynamics of breast cancer relapse reveal late recurring ER-positive genomic subgroups," *Nature*, vol. 486(7403), pp. 346–352, 2012.

[177] N. Aguse, Y. Qi, and M. El-Kebir, "Summarizing the solution space in tumor phylogeny inference by multiple consensus trees," *Bioinformatics*, vol. 35, no. 14, pp. i408–i416, 2019.

[178] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations*, R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, Eds. Springer, 1972, pp. 85–103.

[179] P. Razavi, M. T. Chang, G. Xu, C. Bandlamudi, D. S. Ross, N. Vasan, Y. Cai, C. M. Bielski, M. T. A. Donoghue, P. Jonsson, A. Penson, R. Shen, F. Pareja, R. Kundra, S. Middha, M. L. Cheng, A. Zehir, C. Kandoth, R. Patel, K. Huberman, L. M. Smyth, K. Jhaveri, S. Modi, T. A. Traina, C. Dang, W. Zhang, B. Weigelt, B. T. Li, M. Ladanyi, D. M. Hyman, N. Schultz, M. E. Robson, C. Hudis, E. Brogi, A. Viale, L. Norton, M. N. Dickler, M. F. Berger, C. A. Iacobuzio-Donahue, S. Chandarlapaty, M. Scaltriti, J. S. Reis-Filho, D. B. Solit, B. S. Taylor, and J. Baselga, "The genomic landscape of endocrine-resistant advanced breast cancers," *Cancer Cell*, vol. 34, no. 3, pp. 427–438.e6, 2018.

[180] H. Prüfer, "Neuer beweis eines satzes uber permutationen," *Arch Math Phys*, vol. 27, pp. 742–4, 1918.

[181] P. A. VanderLaan, D. Rangachari, S. M. Mockus, V. Spotlow, H. V. Reddi, J. Malcolm, M. S. Huberman, L. J. Joseph, S. S. Kobayashi, and D. B. Costa, "Mutations in TP53, PIK3CA, PTEN and other genes in EGFR mutated lung cancers: Correlation with clinical outcomes," *Lung Cancer*, vol. 106, pp. 17–21, 2017.