
# Using Trees to Capture Reticulate Evolution

Lateral Gene Transfers and Cancer Progression

ALI TOFIGH

Doctoral Thesis
Stockholm, Sweden, 2009

TRITA-CSC-A 2009:10
ISSN-1653-5723
ISRN–KTH/CSC/A–09/10-SE
ISBN 978-91-7415-349-1

KTH School of Computer Science
and Communication
SE-100 44 Stockholm
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framläggs till offentlig granskning för avläggande av teknologie doktorsexamen i datalogi fredagen den 12 juni 2009 klockan 10.00 i Svedbergssalen, Albanova, Roslagstullsbacken 21, Stockholm.



Tryck: Universitetsservice US AB

*To Bita*

**Abstract**

The historic relationship of species and genes are traditionally depicted using trees. However, not all evolutionary histories are adequately captured by bifurcating processes and an increasing amount of research is devoted towards using networks or network-like structures to capture evolutionary history. Lateral gene transfer (LGT) is a previously controversial mechanism responsible for non tree-like evolutionary histories, and is today accepted as a major force of evolution, particularly in the prokaryotic domain.

In this thesis, we present models of gene evolution incorporating both LGTs and duplications, together with efficient computational methods for various inference problems. Specifically, we define a biologically sound combinatorial model for reconciliation of species and gene trees that facilitates simultaneous consideration of duplications and LGTs. We prove that finding most parsimonious reconciliations is NP-hard, but that the problem can be solved efficiently if reconciliations are not required to be acyclic—a condition that is satisfied when analyzing most real-world datasets. We also provide a polynomial-time algorithm for parametric tree reconciliation, a problem analogous to parametric sequence alignment, that enables us to study the entire space of optimal reconciliations under all possible cost schemes.

Going beyond combinatorial models, we define the first probabilistic model of gene evolution incorporating a birth-death process generating duplications, LGTs, and losses, together with a relaxed molecular clock model of sequence evolution. Algorithms based on Markov chain Monte Carlo (MCMC) techniques, methods from numerical analysis, and dynamic programming are presented for various probability and parameter inference problems.

Finally, we develop methods for analysis of cancer progression, a biological process with many similarities to the process of evolution. Cancer progresses by accumulation of harmful genetic aberrations whose patterns of emergence are graph-like. We develop a model of cancer progression based on trees, and mixtures thereof, that admits an efficient structural EM algorithm for finding Maximum Likelihood (ML) solutions from available cross-sectional data.

# Contents

# List of Publications

- **Simultaneous Identification of Duplications and Lateral Gene Transfers**
  *A. Tofigh, M. Hallett, and J. Lagergren*
  *Based on [67], which was presented at RECOMB 2004*
  *submitted*

- **Inferring Duplications and Lateral Gene Transfers— An Algorithm for Parametric Tree Reconciliation**
  *A. Tofigh and J. Lagergren*
  *Manuscript*

- **Detecting LGTs Using a Novel Probabilistic Model Integrating Duplication, LGTs, Losses, Rate Variation, and Sequence Evolution**
  *A. Tofigh, J. Sjöstrand, B. Sennblad, L. Arvestad, and J. Lagergren*
  *A. Tofigh and J. Sjöstrand have contributed equally to this manuscript*
  *Manuscript*

- **A Global Structural EM algorithm for a Model of Cancer Progression**
  *A. Tofigh and J. Lagergren*
  *Manuscript*

# Acknowledgments

The journey has been long and laborious, but nonetheless quite memorable. I am grateful to all those who provided support and made the tough times less burdensome, and special thanks go to the people who contributed, both directly and indirectly, to the work presented in this thesis.

Foremost among the latter is my supervisor **Jens Lagergren**. He is truly one of the coolest professors I have met and a true scientist—*never* stressed about research, *always* stressed about administration and teaching duties. He is also one of the fastest thinkers I know. It is only through years of training provided by myself and other students that he has learned to come down to the level of us mere mortals when explaining his scientific ideas. I'm especially grateful for his constant support and calm attitude in the face of impending disasters!

Others who have made contributions to this thesis include **Joel**, who has worked hard during weekends to finish experiments and graphs, and **Lasse** and **Bengt** with whom I've had many scientific discussions.

During the years as a PhD student I have shared office space with some very interesting personalities. **Örjan**, my brother in arms, who managed to write his thesis *and* become a first-time father in a matter of just a few months. A true Swedish hero! **Johannes**, is there *anyone* you don't know or haven't met? **Isaac**, it takes sharp minds and brainy people like yourself to build a search engine whose name is now an official verb. **Marcus**, we never shared office space, but it feels like we did. Finally, there was **Samuel**, who left Stockholm and never looked back.

I'm also glad that I got to meet Jens's new PhD students, **Joel** and **Hossein**. Somehow, the blend of Hossein's evilness, Joel's enthusaism, and my lust for revenge led to one of the most successful practical jokes at our department (sorry Jens, but your reaction was priceless!).

A source of welcome distraction has been the (ir)regular poker nights, a tradition I hope will continue for a long time to come. In short, it has been a pleasure losing to you all: **Håkan, Marcus, Katarina, Per, Pär, Diana, Anna, Andreas, Kristoffer, Jenny, Tomas, Björn, Aron, Maria, Erik Sj**.

Finally, there is the one person without whom I would not have survived through it all, and who has had to put up with my shifting moods, especially towards the end: my lovely wife **Bita**, you are too good to be true!

*There is grandeur in this view of life, with its several powers, having been originally breathed into a few forms or into one; and that, whilst this planet has gone cycling on according to the fixed law of gravity, from so simple a beginning endless forms most beautiful and most wonderful have been, and are being, evolved.*

— Charles Darwin,
*On the Origin of Species*

# Chapter 1

# Introduction

Darwin's observations during his five-year voyage on HMS Beagle laid the foundation for the body of work that he would later produce, providing compelling evidence for a process of descent with modification and natural selection. Although evolutionary ideas had been formulated in various forms before Darwin, his work popularized the idea among scientists as well as the general public. Of course, Darwin could not have known that long strands of DNA molecules, and the genes located on them, are the vehicles on which traits are inherited from generation to generation, or that discrete events such as recombination, mutation, duplication, and gene transfer are responsible for modification.

Much of Darwin's observations were based on morphological similarities and dissimilarities between species. In fact, not long ago, classification of species and inference of their evolutionary histories were mainly based on morphological data. With the emergence of modern molecular biology, we have at our disposal a detailed model of the mechanisms of inheritance, and so the study of evolution has largely shifted towards the study of DNA and genes. Methods for the construction of gene trees showing the relationship among homologous genes have been particularly useful. Trees or networks showing the evolutionary history of species are often inferred from sets of gene trees. This problem would be trivial if the history of genes simply followed that of species (or vice versa!), but genome content, and with it, the history of genes, is constantly changing in ways that do not always reflect the history of species. Prominent among these events is gene duplication, which has been extensively studied both biologically and computationally. Another event, one that has risen to prominence more recently, is lateral gene transfer (LGT), also known as horizontal gene transfer (HGT). Many computational methods have been developed that incorporate either duplications or LGTs, but few have attempted to incorporate both. A major part of this thesis concerns the development of computational models and methods for the simultaneous inference of duplications and LGTs.

An evolutionary process on a much smaller scale is seen in cancer progression.

Cancer progresses via accumulation of genetic changes, and evolutionary mechanisms such as selection, competition, predation, and genetic drift characterize this process. The order in which different genetic changes appear during progression of the disease varies between distinct types of cancer. Although cancer progression is best described using graphs or networks, construction of such networks from available data is quite difficult. The last part of this thesis is concerned with a model of cancer progression based on trees, and mixtures thereof, for which we are able to develop efficient algorithms.

The outline of the rest of this thesis is as follows. Chapter 2 provides biological background on evolution and cancer progression. Also, a discussion on the use of trees and methods for tree reconstruction is provided. In Chapter 3, we introduce the major computational techniques that our methods are based on. Chapter 4 describes problems and methods for inference of duplications and lateral gene transfers. Section 4.5 describes the combinatorial model of gene evolution presented in Papers I and II, and Section 4.6 describes the probabilistic model of gene evolution presented in Paper III. Finally, Chapter 5 describes the methods previously developed for construction of cancer progression pathways, and the model and algorithms presented in Paper IV are discussed in Section 5.2. For a brief description of the articles included in this thesis, see Chapter 6.

*A hole had just appeared in the Galaxy . . . Somewhere in the deeply remote past it seriously traumatized a small random group of atoms drifting through the empty sterility of space and made them cling together in the most extraordinarily unlikely patterns. These patterns quickly learnt to copy themselves (this was part of what was so extraordinary about the patterns) and went on to cause massive trouble on every planet they drifted on to. That was how life began in the Universe.*

— Douglas Adams,
*The Hitchhiker's Guide to the Galaxy*

# Chapter 2

# Evolution and Describing its History

The underlying theme of the work presented in this thesis is molecular evolution and computational methods for its study. In this chapter we will give a brief background on and overview of molecular evolution in general, followed by a more detailed discussion of the evolutionary events at the focus of this thesis.

The outline of this chapter is as follows. Sections 2.1 and 2.2 give a brief overview of molecular evolution. In Section 2.5, we will discuss two important evolutionary events that have made a substantial impact in the genetic composition of organisms and which constitute a major focus of this thesis, namely, gene duplications and lateral gene transfers. The process of speciation, i.e., the evolutionary process in which new species emerge from existing ones, is discussed in Section 2.3. Trees have long been used as tools to depict the evolutionary history of organisms. Today they are also used to depict the history of genes that share a common ancestry, so-called homologous genes. A brief discussion of the use of trees and methods for their construction is given in Section 2.4.

A seemingly different, yet closely related, subject is that of cancer progression. "Cancer" is a name that refers to a large class of diseases with uncontrolled cell growth and proliferation as a common characteristic. The abnormal properties of cancerous cells are due to accumulation of harmful genetic changes. This process, called *somatic* evolution, is very similar to evolution of species and is discussed in Section 2.6.

## 2.1 Introduction to Genetics and Genomics

The observation that offspring inherit traits from their parents has long been used by humans, e.g., in breeding of animals and plants. Gregor Mendel performed the first systematic study of the basis of inheritance for some simple discrete traits, such as the color of the flower of the common pea plant [107, 108]. His discoveries

concerning dominant and recessive traits became known as Mendelian inheritance.

Mendel's work did not receive any significant attention until it was rediscovered in the beginning of the 20th century. Research then led to the discovery that genes, the basic functional units of heredity, reside on the chromosomes, a discovery that was awarded with the Nobel prize in 1933. Although chromosomes were identified as the carriers of genetic material, the composition of the genetic material was yet unknown. The first experiments showing that the genetic information was contained in the DNA of chromosomes were performed by Avery *et al.* [8] and was later confirmed by Hershey and Chase [77].

James D. Watson and Francis Crick published the first accurate model of DNA structure in 1953 [150], and the genetic code was cracked by Har Gobind Khorana, Robert W. Holley, and Marshall Nirenberg, who shared the Nobel prize in physiology in 1968.

The central dogma of molecular biology, that the flow of information in the cell goes from DNA to mRNA, to protein, but never from protein to nucleic acid, was formulated by Francis Crick [56, 31].

Today, technological advances have enabled us to sequence entire genomes. All 6 billion bases comprising the genome of James D. Watson were sequenced in two months time in 2008 [152]. One of the challenges of the future lies in constructing computational tools for extracting functional information from sequence data. In the following, we give a brief overview of the molecular machinery of the cell responsible for reading the genome and producing the proteins that are responsible for most of a cell's functions. For more in-depth information, we refer to the standard text book by Bruce Alberts *et al.* [1].

### The Central Dogma of Molecular Biology

The cell is the smallest structural and functional unit of an organism that is classified as living. There are two types of cells: eukaryotes comprising multicellular animals, plants, fungi, as well as unicellular organisms, and prokaryotes such as bacteria. "Karyose" comes from a Greek word meaning kernel, "pro" means before, and "eu" means true. So prokaryotic means "before a nucleus", and eukaryotic means "possessing a true nucleus". The name emphasizes the fact that eukaryotes carry their genetic material inside a cell nucleus, while prokaryotes have no such compartment and the genetic material is held within the cytosol. The genetic material of both eukaryotes and prokaryotes consists of long molecules of deoxyribonucleic acid (DNA). As the eukaryotic DNA molecules are very long and have to fit in a small nucleus, they are folded up into chromosomes in a highly organized manner. The prokaryotic DNA, on the other hand, is present as circular naked DNA molecules. DNA acts like an instruction manual and its sequence provides all the information needed for a cell to function. The information is first copied to ribonucleic acid (RNA) before being transformed into proteins—this is the so-called dogma of molecular biology, namely that information passes from nucleotides to amino acids but never in the opposite direction. The functional units in the DNA

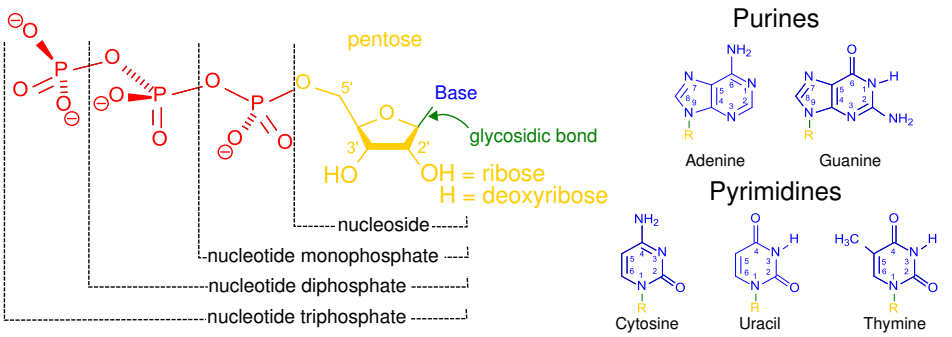



Figure 2.1: ***The structure of nucleotides.***

that code for RNA or proteins are called genes. Each gene encodes one or a set of similar proteins, and each protein performs a specialized function in the cell. Cells use the two-step process of transcription and translation to read genes and produce the strings of amino acids that make up a protein. The production of the various proteins is one of the most important processes occurring inside a cell as proteins not only form structural components of the cell, but they also compose the enzymes that catalyze the production of other organic biomolecules required for the cell to function.

## DNA and RNA Structure

DNA is the carrier of genetic information composed of four different nucleotides. Each nucleotide is composed of three parts: (1) a nitrogenous base known as purine (adenine (A) and guanine (G)) or pyrimidine (cytosine (C) and thymine (T)); (2) a sugar, deoxyribose; and (3) a phosphate group. The nitrogenous base determines the identity of the nucleotide, and individual nucleotides are often referred to by their base (A, C, G, or T), see Figure 2.1. One DNA strand can consist of up to several hundred million nucleotides. The nucleotide T can form a hydrogen bond with A, and C with G, making a double-helix formed by two anti-parallel complementary strands, see Figure 2.2.

RNA is very similar to DNA, the only difference being that the pyrimidine base thymine is replaced by uracil (U) and the ribose comes in its fully hydroxylated form. Together, the presence of uracil in place of thymine, and the 2′-OH in the ribose constitute the two chemical differences between RNA and DNA. Also, RNA does not form a double helical structure and is in general single-stranded. There are many types of RNA present in the cell distinguished by their functional role. Three of these, namely messenger RNA (mRNA), transfer RNA (tRNA), and ribosomal RNA (rRNA), will be discussed in further detail below.

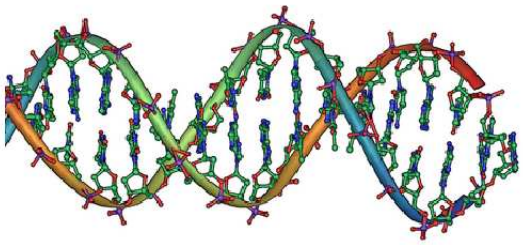


Figure 2.2: ***DNA double helix.***

### Transcription

Transcription refers to the transfer of the genetic code from DNA to a complementary RNA and occurs in the cell nucleus. The mRNA serves as an intermediate between DNA and protein. The transcription starts with the enzyme RNA polymerase attaching and unzipping the DNA molecule into two separate strands after which it binds to the promoter segment of DNA that indicates the beginning of the single strand of DNA to be copied. It moves along the DNA and matches each DNA nucleotide with a complementary RNA nucleotide to create a new RNA molecule patterned after the DNA. The copying of the DNA continues until RNA polymerase reaches a termination signal, i.e., a specific set of nucleotides that mark the end of the gene to be copied. When the RNA polymerase has finished copying a particular segment of DNA, the DNA reconfigures into the original double-helix structure. In prokaryotes, this RNA needs no further processing and provides the blueprint which directs protein synthesis. However, in eukaryotes, this RNA strand (the transcript) is first processed into mature mRNA. The processing involves the removal of intervening non-coding sequences, so-called introns. The newly created mRNA is then exported out of the nucleus and into the cytoplasm where translation can take place.

### Translation

Translation refers to the process of converting the information contained in an mRNA molecule into a sequence of amino acids that bind together to form a protein. In the cytosol, mRNA molecules bind to protein-RNA complexes called ribosomes. Each ribosome includes a large and a small subunit containing rRNA and more than 50 proteins. The small and large subunits of the ribosome surround the mRNA after which tRNA molecules carrying amino acids attach to the ribosome and mRNA to create the polypeptide chain, see Figure 2.3. There are several types of tRNA molecules each, containing a unique three base region called the anticodon that can base pair to the corresponding three base codon on the mRNA. Each type of tRNA

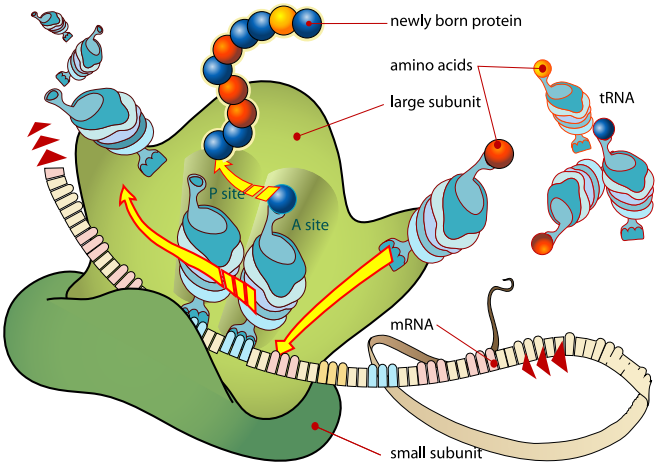



Figure 2.3: ***A schematic view of translation.***

molecule can only carry one unique amino acid, but may base pair to more than one codon sequence on the mRNA. Hence, each three base codon signals for the inclusion of a specific amino acid, but the same amino acid can be coded by several different codons. The correspondence between codons and amino acids is called the genetic code and is shown in Figure 2.4.

## 2.2 Genome Evolution

About 3.5 billion years ago, cells similar to modern day bacteria had appeared. There is evidence for the existence of eukaryotic cells 1.4 billion years ago with the first multicellular animals appearing around 640 million years ago [21]. In this section, we provide a few examples of the diverse (molecular) evolutionary events that have shaped present day genomes during millions of years of evolution.

Although cells have acquired highly complex and accurate mechanisms for DNA replication and repair, a cell can still fail to create exact copies of its chromosomal DNA during cell division. In fact, such failures are the predominant causes of genetic changes during evolution, although transposable DNA elements also play a major role.

In the context of this thesis we assume that there exists a reference genome that is representative of the genome of all individuals belonging to a certain species. For now, we also assume that the concept of species and the classification of individuals as belonging to one species is unproblematic, though as we will see, this has been contested in recent years mainly withing the prokaryotic domain.

We can imagine following the fate of a single gene as it is passed from one gen-

| 1st Position | 2nd Position: U | 2nd Position: C | 2nd Position: A | 2nd Position: G | 3rd Position |
|---|---|---|---|---|---|
| **U** | Phe | Ser | Tyr | Cys | **U** |
| | Phe | Ser | Tyr | Cys | **C** |
| | Leu | Ser | * | * | **A** |
| | Leu | Ser | * | Trp | **G** |
| **C** | Leu | Pro | His | Arg | **U** |
| | Leu | Pro | His | Arg | **C** |
| | Leu | Pro | Gln | Arg | **A** |
| | Leu | Pro | Gln | Arg | **G** |
| **A** | Ile | Thr | Asn | Ser | **U** |
| | Ile | Thr | Asn | Ser | **C** |
| | Ile | Thr | Lys | Arg | **A** |
| | Met | Thr | Lys | Arg | **G** |
| **G** | Val | Ala | Asp | Gly | **U** |
| | Val | Ala | Asp | Gly | **C** |
| | Val | Ala | Glu | Gly | **A** |
| | Val | Ala | Glu | Gly | **G** |

Figure 2.4: ***The genetic code.*** *A * indicates stop-codon.*

eration to the next in one species. For various reasons, we would observe changes to the DNA sequence of the gene. On a small scale, we would detect substitutions, insertions, and deletions involving a few nucleotides. Translocations, exon duplications, exon shuffling, and gene conversion are examples of other small scale events that alter the sequence of a gene.

Mathematical models of sequence evolution, so-called substitution models, have been proposed and are routinely used, e.g., to reconstruct trees showing the relationship between genes from different organisms. These models together with tree reconstruction methods are discussed in Section 2.4.

Taking genes to be atomic units, i.e., ignoring changes to the DNA sequence, we would see larger scale events that affect entire genes. For example, portions of chromosomes are sometimes duplicated or deleted and may affect a whole set of genes that reside on those segments. We would see how genes are lost, for example due to segmental deletions that remove the sequence or a part of the sequence from the genome altogether, or deleterious mutations that cause the silencing of the gene. The number of genes can also increase via events such as gene duplications (caused, e.g., by segmental duplications or reverse transcription), lateral gene transfers (the transfer of genetic material from one species or individual to another), or interstitial deletions (segmental deletions that do not include chromosomal endpoints; these may cause two genes to be fused together into one gene). On the largest scale, we have whole genome duplications that doubles the number of chromosomes and

hence also the number of genes in a species, but is usually followed by massive gene losses.

The work presented in this thesis deals mainly with two of the evolutionary events mentioned above, namely gene duplications and lateral gene transfers. These are therefore discussed in more detail later in Section 2.5.

## 2.3 Speciations and Organismal Trees

Ever since Darwin's work popularized the idea of evolution, trees have been widely used to depict the historic relationship between species. When groups of individuals belonging to the same species are isolated from each other, they are independently affected by the evolutionary processes. Over time the groups will form distinct species, an event that we call speciation. Trees are often the best representation of the process of speciation in higher organism, although plants and fish are known to hybridize to form new species. In these cases, the speciation process is best represented by networks.

The classification of micro-organisms into species can sometimes be problematic. For example, it has become increasingly apparent that lateral gene transfers have played a major role in prokaryotic evolution, and some have argued that a species tree representing prokaryotic evolution, at least the evolution of certain groups of taxa, may not exist. Others argue that although lateral gene transfers have played a major role, the notion of species and species trees are still meaningful representations of the evolutionary history of prokaryotes.

## 2.4 Phylogeny and Tree Reconstruction

Before the emergence of the modern theory of molecular evolution, classification of species and inference of phylogenies were based on morphological data. With the modern understanding of the molecular mechanisms of inheritance, practice has shifted to using DNA or amino acid sequence data as the basis for reconstructing evolutionary histories. The history of a set of homologous genes is usually adequately represented by a tree, although there are some events such as gene conversion and recombination that are responsible for creating non tree-like histories. The history of species, whether represented by trees or networks, is often inferred from a set of gene trees. In this section we will discuss some of the more popular methods for gene tree reconstruction.

Computational methods for tree reconstruction attempt to find a tree or a set of trees that are optimal according to some criteria. Several different criteria have been used when devising computational methods for tree reconstruction. A number of methods have been proposed based on parsimony where we seek the tree that requires a minimum number of evolutionary events to explain the data (see Section 3.1 for a general discussion of parsimony). The first to suggest the use of parsimony as a criterion for tree reconstruction were Edwards and Cavalli-Sforza [45].

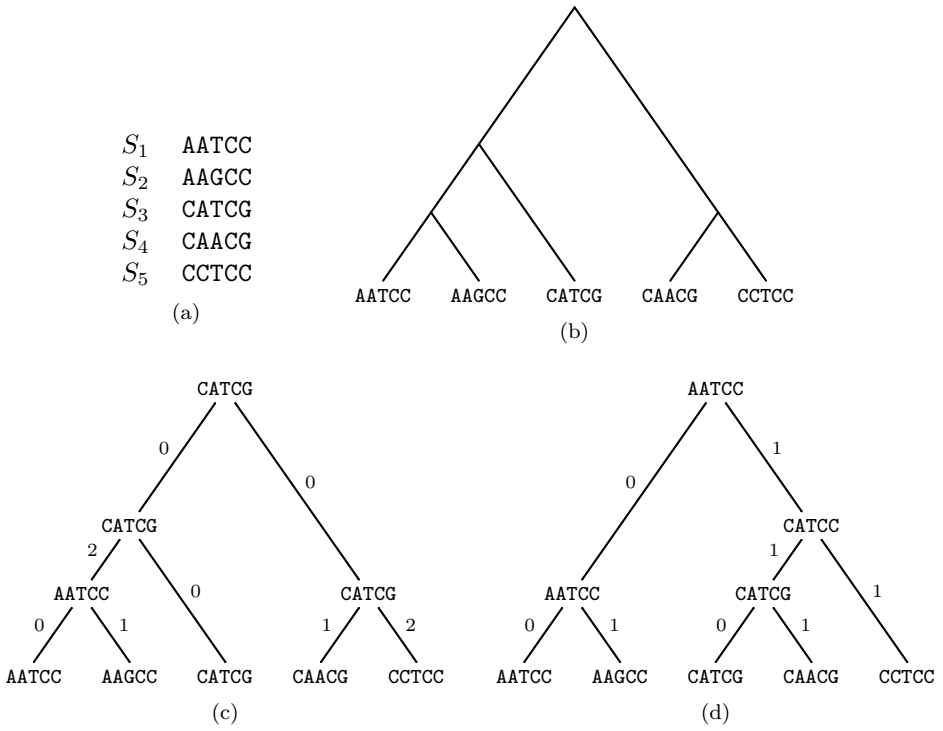



Figure 2.5: ***Maximum parsimony example****. (a) shows a set of five DNA sequences for which we seek a parsimony tree. (b) shows an arbitrary tree whose leaves are associated with the sequences. (c) shows one way to assign hypothetical sequences to the internal vertices to the tree in (b) such that the total number of substitutions is minimized. For this tree, at least 6 substitutions are required. (d) shows a maximum parsimony tree with hypothetical sequences assigned to internal vertices. This tree requires only 5 substitutions which is the least number of substitutions required for any tree and any assignment of sequences to internal vertices.*

Assume that we are given $n$ sequences. For each rooted bifurcating tree with $n$ leaves corresponding to the $n$ sequences and an assignment of hypothetical sequences to the internal vertices, we can compute the minimal sets of evolutionary events that have taken place along each tree edge. Traditionally, an evolutionary event is defined as a single substitution of one nucleotide or amino acid for another. If we assign a cost to each possible substitution, we can compute a cost for each tree and assignment. The cost of a tree is then defined as the minimum cost over all possible assignments of sequences to its internal vertices. Out task is now to find the tree with the least cost. See Figure 2.5 for a complete example. A general algorithm for computing the minimal cost of a given tree was given by Sankoff [137] and Sankoff and Rousseau [138]. Finding the best tree can be done by searching in the space of trees using different heuristics and local search algorithms.

The criteria given above is quite general and various special cases together with specialized algorithms have been described in the literature. Other variants of parsimony exist that are not easily solved using the Sankoff algorithm, for example polymorphism parsimony [48, 51].

Distance matrix methods comprise another major class of phylogenetic methods. These methods were introduced in [25] and [58]. We can associate with each edge of a phylogenetic tree a branch length representing the total amount of evolution that has occurred between the two vertices. By summing the branch lengths of all the edges on the path between two leaves, we can compute pairwise distances between each pair of sequences or species. Distance matrix methods take as input pairwise distances between sequences and attempt to reconstruct a phylogenetic tree with branch lengths such that the distances induced by the tree is as close as possible to the given distances.

In order to develop algorithms for the problem, we need to define precisely how to measure closeness between the given pairwise distances and those induced by a tree. A popular, and statistically justifiable, criterion is *least squares*. Let $D_{ij}$ be the given distance between sequence $i$ and $j$ and let $D_{ij}^T$ be the distance as induced by a tree $T$. The best tree according to the least squares criterion is a tree $T$ that minimizes the following expression:

$$\sum_{ij}(D_{ij} - D_{ij}^T)^2. \tag{2.1}$$

There are efficient algorithms for determining branch lengths that minimize (2.1) given a fixed tree [63]. Finding the optimal tree according to the least squares criterion is, however, NP-hard [35]. Searching for a good tree is usually done by heuristics and local search methods.

Another criterion used by distance matrix methods is that of minimum evolution. In the minimum evolution methods, the branch lengths of a tree are determined by the least squares criterion, but the optimality criterion used to choose between trees is different. Instead of choosing the tree whose induced distances is as close as possible to the given distances, the tree with the minimum total length is preferred [87, 133].

There are also other heuristic distance matrix methods that do not have an explicit optimality criterion. One of the most popular methods for tree reconstruction is called neighbor-joining (NJ) [134] and is based on the clustering algorithms popularized by Sokal and Sneath [142]. Although NJ is defined by its algorithmic description, it is related to both the minimum evolution and least squares criteria. An interesting property of NJ is that it will reconstruct the correct tree if the given distances are "sufficiently close" to the distances induced by the tree. More formally, a distance matrix $D$ is said to be *nearly additive* if there is a tree $T$ with induced distance matrix $D^T$ such that

$$|D - D_{ij}^T|_\infty < \frac{\mu(T)}{2},$$

where $\mu(T)$ is the minimum edge length in $T$. It can be shown that given a nearly additive distance matrix, NJ will reconstruct the unique tree $T$ [6]. There are several other methods with this property that do not work as well in practice. However, FastNJ [47] is an algorithm that is very similar to NJ, works well in practice, and is much more efficient.

Distance matrix methods require calculations of pairwise distances of sequences. Due to insertions and deletions, homologous sequences are almost always of different lengths and the homologous positions are not immediately apparent. Hence, sequences must be *aligned* before computing distances. Many different exact and heuristic methods exist for sequence alignment and a proper discussion is beyond the scope of this thesis. We refer instead to several recent reviews on the state of modern alignment algorithms [91, 44, 123]. In general, alignment algorithms attempt to produce rows of sequences with inserted gaps such that the nucleotides or amino acids in each column are homologous. Often, columns that include gaps are discarded and distances are based only on columns without gaps.

Given an alignment, the naive approach to computing a distance between a pair of sequences is to count the number of mismatches. This leads, however, to an underestimation of the amount of substitutions that have occurred since more than one substitution may be responsible for a single mismatch or even a match since substitutions can be reversed by subsequent mutations. A better approach is instead to use a probabilistic model of sequence evolution to estimate the number of substitutions. Several substitution models have been proposed for DNA and amino acid sequences with varying amounts of complexity. Models of DNA evolution include Jukes-Cantor [84], Kimura's two-parameter model [88], F84 [54, 89], HKY [72], and the Tamura-Nei model [144].

Jukes-Cantor is the simplest model and assumes that substitutions occur according to a Poisson process with rate $\alpha$ and that all substitutions are equally likely. For this simple model, there are closed formulas for calculating the maximum likelihood estimate of the number of mutations. More complex models allow different rates to be assigned to different types of substitutions, for example transitions (substitution of one purine for another or of one pyrimidine for another) and transversions (substitution of a purine for a pyrimidine or vice versa). Numerical methods are used for the more complex models to obtain maximum likelihood estimates of distances or branch lengths. Substitution models exist also for protein sequences though we will not discuss them here.

Instead of using the substitution models described above to compute distances and then use distance methods to obtain a tree, it is possible to use the models and sequences more directly. Sequences together with a substitution model induce a probability distribution on trees with branch lengths. Hence, a natural problem is to find the maximum likelihood tree, i.e., the tree which maximizes the probability of observing the sequences given the substitution model. Phylogenetic likelihood methods were popularized by Felsenstein, see for example [52]. See also Section 3.3 which discusses structural EM algorithms.

Recent years have seen a growing body of Bayesian methods being developed.

Bayesian statistical inference is discussed in Section 3.4. A very popular computer program for Bayesian inference of phylogenetic trees is MrBayes [78, 132]. For an excellent introduction to phylogeny inference in general, see [53]. Felsenstein maintains a comprehensive list of phylogeny software that can be found at http://evolution.gs.washington.edu/phylip/software.html.

The discussion above has centered around construction of trees from sequence data. We often find that trees constructed from sequences of different genes from the same set of organisms are not identical. Species trees can be constructed based on sets of core genes, such as those involved in the transcription or translation machinery, whose evolution is believed to closely follow that of the corresponding species. Other methods include tree consensus methods that are based on the collective signal from a large set of gene trees, and the use of concatenated sequences from many different genes. For some groups of organisms, such as humans, apes, and rodents, we may have other kinds of information, such as archaeological data, available that can also be used for estimation of organismal phylogenies.

Irrespective of our method of choice, we are bound to observe that gene trees and species tree are different and that the evolutionary history of genes do not always follow that of the corresponding species. This poses the problem of reconciling the differences between trees by identifying the responsible evolutionary events. Tree reconciliation is a major part of this thesis and will be discussed in coming sections and chapters.

## 2.5 Gene Duplications and Lateral Gene Transfers

In this chapter, we will take a deeper look at the two evolutionary events with which this thesis is mainly concerned. The importance of gene duplication and its role as a major driving force of evolution has been established. This is in contrast to the role of lateral gene transfer (LGT) which has been the subject of much controversy. The next two subsections will deal with gene duplications and the controversy surrounding LGTs.

### Gene Duplications

The role of gene duplication as a major driving force of evolution has been recognized for a long time. Ohno's seminal book *Evolution by Gene Duplication* [127] in 1970 popularized the idea among biologists, although it had been discussed and debated much earlier. For example, already in 1918, Bridges speculated that duplicate genes can mutate separately thus diversifying their functions [18]. See also later papers by Bridges [19] and Muller [117, 118]. For a review of the history of these ideas, see [145].

Several mechanisms are responsible for creating copies of a gene. These include unequal crossing-over, retrotransposition, segmental duplication, and whole genome duplication.

Unequal crossing-over occurs when homologous chromosomes are not precisely paired during recombination and results in chromosomes of unequal length: one chromosome acquires more genetic material than it passes over and thus contains a duplicated segment. This segment may contain part of a gene, an entire gene, or several genes. Genes duplicated via unequal crossing-over are located on the same chromosome, at least initially, before other events change their relative locations.

Retrotransposition is the process during which a messenger RNA (mRNA) is retrotranscribed to copy DNA (cDNA) and is then inserted into the genome, probably at a random location on some chromosome. The two versions of the gene generally reside on different chromosomes, and the copy also lacks introns since the introns have been spliced out before the mRNA is copied to cDNA.

Segmental duplications, i.e., duplications of large segments of a chromosome, are also responsible for duplication of genes and have been shown to have occurred frequently during primate evolution. The sizes of the duplicated segments tend to be somewhere between 1 000 to 200 000 nucleotides [135, 106]. The exact mechanisms creating such duplications are not clear though several models have been proposed [10].

Whole genome duplications, which may occur during abnormal cell division, is most commonly found in plants, and also in fishes [112]. There is also evidence indicating that one or two whole genome duplications have occurred very early in vertebrate evolution [36]. Whole genome duplications are usually accompanied by massive gene losses [29, 22].

The rate with which gene duplication occurs in eukaryotes has been estimated to one duplication per gene per 100 million years, which is similar to the rate of nucleotide substitutions [100]. Although the rate of duplication is high, most duplications are followed by gene loss. The fates of recently duplicated genes were termed non-functionalization, sub-functionalization, and neo-functionalization in [101, 59].

Analysis of sequenced genomes have revealed that a substantial proportion of genes are duplicated and that the distribution of gene family sizes across species varies greatly [154]. For example, the biggest gene family in *Drosophila melanogaster* has 111 members, while the biggest family in mammals is the olfactory receptor family with more that a thousand members. The KRAB-zinc finger family is another example of a gene family that has undergone many recent gene duplication events and there are over 400 active members of the gene present in the human genome [70].

### Lateral Gene Transfers

Contrary to gene duplications, the importance and prevalence of lateral gene transfers has been much more controversial. LGT refers to the transfer of genetic material from one individual to another. The possibility of LGT in bacteria was realized already in the 1940s [93, 94] and demonstrated to occur between different species in 1959 [125]. We know today that LGT occurs frequently among prokaryotes [126, 20], and that it also occurs from prokaryotes to eukaryotes and among

eukaryotes [85], though probably not as frequently as in the prokaryotic domain.

The abundance of LGTs in prokaryotes has led to some researchers challenging the idea that phylogenetic trees are able to represent prokaryotic evolution, see for example [64, 41, 153], and also [42] and references therein. There is an emerging view today that although LGTs have occurred among prokaryotes, perhaps it has not occurred so much that we must abandon trees altogether [11].

The mechanisms of LGTs among prokaryotes include transformation, transduction, and conjugation. Transformation refers to the uptake of DNA which is then incorporated into the genome. Certain bacteria have a natural ability to take up DNA from their environments. Transduction refers to the process of genetic exchange between bacteria mediated by a bacterial virus, a bacteriophage. Conjugation is a process in which bacterial cells transfer genetic material via direct contact.

Lateral gene transfer has also become an important medical issue [143] as it plays a major role in the spread of antibiotic resistance genes among pathogenic bacteria. Recently, the role of LGT in pathogen evolution has received much attention [92].

## 2.6 Progression in Cancer—an Evolutionary Phenomenon

Cancer is the name given to a whole host of genetic diseases in which cells undergo uncontrolled growth. Genetic alterations to three types of genes are responsible for tumorigenesis—the process in which normal cells are transformed into cancer cells. These are the gatekeepers, caretakers, and landscapers [113]. Gatekeepers are genes that directly affect growth and differentiation of cells and include the oncogenes and tumor suppressor genes, i.e., genes whose abnormal activation and suppression, respectively, can turn normal cells into cancer cells. Caretakers are responsible for maintaining the genomic integrity of cells and promote tumorigenesis indirectly. Alterations to caretaker genes can lead to genetic instability causing rapid accumulation of changes to the genome. Such changes can affect oncogenes or tumor suppressor genes which in turn leads to abnormal proliferation. As the name suggests, landscapers affect the micro-environment of cells. Landscaper genes cause tumorigenesis indirectly by generating an abnormal stromal environment [15]. When the normal intercellular signals are disrupted, for example during sustained inflammation, cells possessing tumorigenic potential can start to proliferate uncontrollably. Such abnormal conditions can also cause genetic instability which then could lead to development of cancer. For example, it has been known for more than a century that inflammation associated with tissue wounding can produce tumors, see [40, 140] and references therein.

Although cancer is a generic name for different diseases, six "hallmarks of cancer" common to all cancer types have been identified [71]. These are self-sufficiency in growth signals, insensitivity to anti-growth signals, apoptosis-evasion (evasion of programmed cell death), limitless replicative potential, sustained angiogenesis (the growth of new blood vessels), and tissue invasion and metastasis. Acquiring all

these traits requires major genetic alterations that are accumulated as cancer progresses towards further malignancy. Although the rate of nucleotide mutation does not appear to be higher in cancer cells [148], chromosomal instability (CIN) seems to be present in all types of human cancer [96]. Mutations in CIN genes increase the rate with which whole chromosomes or large parts of chromosomes are lost or gained during cell division. Aneuploidy, i.e., imbalances in the number of chromosomes, and increased rates of loss of heterozygocity are caused by CIN. In [71], CIN was not identified as a hallmark of cancer but was taken to be a prerequisite for acquiring the entire set of hallmarks.

The progression of cancer and the acquisition of the previously mentioned traits is an evolutionary process involving selection among genetically variable cells [124, 30]. The evolution of cells within the body is called somatic evolution. Somatic evolution shares many similarities with evolution of organisms and many methods and models from population genetics [113] and ecology [30] can be applied to cancer progression, although some differences do exist and models may have to be altered to take these into consideration. Important factors that play major roles in somatic evolution include mutation, genetic drift, natural selection, competition, predation, and dispersal or colonization. A neoplasm, or tumor, consists of a large population of genetically heterogeneous individuals [103, 24] that undergo selective sweeps followed by clonal expansions, see for example [105] and [104]. Clones, i.e., a group of cells derived from a single mother cell, can expand or contract based on their fitness in the population. In general, evolution within a tumor population selects for increased growth and survival and mutations can become fixed in the population during selective sweeps.

The rates of mutation in cancer cells remain undetermined *in vivo*, but there are indications that they are not higher than in normal cells. In cell culture, the rates have been determined to be somewhere between $10^6$ and $10^7$ per locus per generation [2]. In [148], the number of non-synonymous mutations in colorectal tumor cells was determined to approximately 1 mutation per megabase of DNA, which is similar to the expected number of mutations in normal cells that have undergone as many generations and population size bottlenecks. The number of mutations required to cause cancer is not precisely known, but is probably somewhere between 3 and 12 depending on the type of cancer [131]. Given the incidence rate of cancer in the human population, it seems unlikely that so many mutations could be accumulated in a single cell based solely on normal somatic mutation rates [99, 71]. One explanation could be that the expansion of clones provides large enough populations to produce subsequent necessary mutations [115].

In some cases mutation in a gene gives its host a selective advantage only after another gene has undergone mutation. For example, in Barrett's Esophagus, the inactivation of *TP53* almost always occurs after *CDKN2A* has been inactivated [105]. It could be the case that inactivation of *CDKN2A* initiates a clonal expansion that provides opportunities for mutations to occur to *TP53* after which a second clonal expansion would occur. Such dependencies provide opportunities for modeling of cancer progression which is discussed in Chapter 5.

Competition and predation are other evolutionary forces that can act on organisms and have analogs in somatic evolution. For example, cells in a tumor are constantly competing for resources. More complex modes of competition have also been shown. For example, clones on different locations in the same mouse or rat can inhibit each other's growth [114, 23]. Aspects similar to predation in ecology is also present in cancer progression, most naturally from the immune system. One difference compared to ecology is that the extinction of prey does not lead to the extinction of the predator.

# Chapter 3

# Computational Techniques

This chapter provides some background on a selection of computational techniques that have been used in the work presented in this thesis. Parsimony can be explained as a general principle of "less is more". Papers I and II deal with methods for finding the "simplest" reconciliations of trees and Section 3.1 provides a brief overview of the subject of parsimony. In Paper I, we also develop a fixed-parameter tractable algorithm for the tree reconciliation problem. Some general comments about parametrized complexity is given in Section 3.2. Expectation Maximization (EM) is an iterative meta algorithm for finding maximum likelihood estimates in probabilistic models. Paper IV of this thesis provides an EM algorithm for a model of cancer progression. A general description of EM algorithms is given in Section 3.3. Bayesian methods have been applied quite successfully on a large set of problems in bioinformatics, though their use is sometimes controversial. In Paper III of this thesis, we develop algorithms and methods for Bayesian inference of duplications and lateral gene transfers in which Markov Chain Monte Carlo (MCMC) techniques play a major role. A brief overview of Bayesian methods and the degree-of-belief interpretation of probability is given in Section 3.4 along with a discussion of Markov Chain Monte Carlo (MCMC) Techniques.

## 3.1 Ockham's Razor and Parsimony

*Numquam ponenda est pluralitas sine necessitate*
William of Ockham

In bioinformatics, maximum parsimony is probably the most well-known example of the use of the principle of Ockham's razor—that "plurality should never be posited without necessity". In [45], Edwards and Cavalli-Sforza first mentioned the general idea of maximum parsimony when they declared that the preferred evolutionary tree is the one that involves the minimum net amount of evolution, and for a long

time, parsimony methods were the most widely used tree reconstruction methods for character data.

The general principle of parsimony is, of course, not restricted to evolutionary trees. In fact, the principle has been advocated many times in the past, even long before Ockham applied it to such an extent as to give it its current name. A famous example in scientific literature is Newton's first rule of reasoning in philosophy as stated in "Mathematical Principles of Natural Philosophy":

> *Rule I We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances.*
>
> *To this purpose the philosophers say that Nature does nothing in vain, and more is in vain when less will serve; for Nature is pleased with simplicity, and affects not the pomp of superfluous causes.*
>
> *RULE II Therefore to the same natural effects we must, as far as possible, assign the same causes.*

In general, parsimony can be applied in two different ways. On the one hand, we can take parsimony as an optimality criterion, as in the case of maximum parsimony in phylogenetics: trees are scored based on the level of complexity with which they can explain the data and the trees which require fewer assumptions of evolutionary events are preferred over trees that require more. On the other hand, parsimony can be applied when designing models in the sense of defining models with no more parameters than necessary to sufficiently model the data. A simple example of this is fitting a curve to points in the plane. In finance, models for prediction of yield curves are another example where parsimony has been explicitly applied when devising models [121, 39]. Many more examples can be found where parsimony is used implicitly, and there is a vast amount of literature in statistics concerning the choice of model and the number of parameters.

In this thesis, parsimony is applied to the problem of tree reconciliation which is explained in Chapter 4. Analogous to tree reconstruction, parsimony is here used as an optimality criterion where the simplest reconciliation is sought in the sense of minimizing the number of evolutionary events needed to explain the differences between an organismal tree and a corresponding gene tree. Early work on tree reconciliation problems sought to reconcile trees using duplications and losses. In this case, defining the underlying combinatorial model is quite straightforward. A level of complexity is added when we also have to consider lateral gene transfers, and care must be taken when defining the combinatorial problem to ensure biological feasibility. The combinatorial model developed in this thesis is biologically feasible and an extensive discussion of this issue can be found in Paper II.

A common objection to the use of parsimony is that nature and evolution are not constrained to being parsimonious. The usual answer is that we are not making an assumption about nature or evolution, and that parsimony methods can yield quite complex solutions. We merely strive to find a minimal set of assumptions needed to explain the data. In any case, the strength of any method lies in its predictive

strength, or in the case of tree reconstruction, the ability to correctly infer the past. Even when the true process that generates the data is highly complex, it may be a good idea to use simple models, at least when the sample size is small.

## 3.2 Parameterized Complexity

Computational complexity theory is a branch of theoretical computer science concerned with analyzing the amount of resources needed to solve computational problems. The most important computational resources are time and memory. In complexity theory, problems are categorized into complexity classes based on the amount of resources needed to solve them. The classes P and NP are the most studied due to their practical implications, and the question $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ is one of the most important unsolved problems in theoretical computer science and mathematics. For an introduction to this topic, we refer the reader to the classic book of Garey and Johnson [62].

The time complexity of an algorithm is measured as a function, say $f$, of the input size. If $n$ is the size of the input, then $f(n)$ is the maximum number of steps that the algorithm needs to produce its output. Under the unit cost model, a step is any basic operation such as addition, multiplication, or comparison. Although there are cases when it is preferable to keep separate counts of different operations, e.g., by analyzing the number of multiplications and additions separately, we will only consider the unit cost model in this text. When comparing the time complexity of different algorithms, we are mainly interested in their asymptotic behavior. We say that a function $f(n)$ is $O(g(n))$, if there is a constant $C$ such that $|f(n)| \leq C \cdot |g(n)|$ for all sufficiently large $n$. A polynomial time algorithm is one whose time complexity is $O(p(n))$ where $p$ is some polynomial. Algorithms with polynomial time complexity are considered efficient and problems for which polynomial time algorithms are known are considered tractable. This definition of computational efficiency has proved extremely successful when dealing with natural or real-world problems; in the vast majority of real-world cases, polynomial time algorithms are sufficiently efficient.

When confronted with a problem that does not seem to admit a polynomial time solution, the traditional way to deal with it is to try to show that the problem is NP-hard. The theory of P and NP deals only with decision problems. As an example, take the traveling salesman problem in which we are given a set of cities together with distances between each pair. We want to find a minimal length tour that visits all cities. This optimization problem can be recast as a decision problem: given the cities and the distances between them, is there a tour that visits all cities and whose total length is at most $K$? Clearly, the optimization problem is at least as hard as the decision problem, and so, if the decision problem can be shown to be hard, then the optimization problem must also be hard. The complexity class P consists of all decision problems that admit a polynomial time algorithm. The class NP consists of all decision problems whose "yes"-instances can be verified in

polynomial time. By *verifying the "yes"-instances*, we mean that for each "yes"-instance, there is a certificate with the help of which we can check that the instance really is a "yes"-instance. For example, in the case of the traveling salesman, a certificate consists of a minimal length tour, and checking that a tour visits all cities and has length no more than $K$ can be done in polynomial time. Hence, traveling salesman is in NP. In fact, it is one of the most well-known examples of NP-complete problems. From the definition of P and NP, it is clear that P is a subset of NP. Most researchers believe that the converse is not true, although no proof of this fact has been found.

One benefit of studying decision problems is that we can describe them in terms of the formal notion of languages. For any finite non-empty set of symbols $\Sigma$, let $\Sigma^*$ denote the set of all strings of symbols from $\Sigma$. A set $L$ is a language over the alphabet $\Sigma$ if it is a subset of $\Sigma^*$. The instances of a decision problem can always be encoded by strings of symbols from $\Sigma$, e.g., when $\Sigma = \{0, 1\}$. The language corresponding to a decision problem $\mathcal{P}$ is simply the subset $L_{\mathcal{P}} \subseteq \Sigma^*$ whose members are the encoded "yes"-instances of $\mathcal{P}$. We say that an algorithm decides a language $L$, if it returns "yes" when presented with an element of $L$ and "no" otherwise. A problem $\mathcal{P}$ is in the class P if there is a polynomial time algorithm that decides $L_{\mathcal{P}}$.

A problem is NP-complete if every problem in NP can be reduced to it via a polynomial time algorithm. We say that a problem $\mathcal{P}_1$ can be reduced to problem $\mathcal{P}_2$ if there is a polynomial time algorithm that transforms each instance $x_1$ of $\mathcal{P}_1$ into an instance $x_2$ of $\mathcal{P}_2$ such that $x_1$ is a "yes"-instance of $\mathcal{P}_1$ if and only if $x_2$ is a "yes"-instance of $\mathcal{P}_2$. Clearly, if $\mathcal{P}_2$ can be solved in polynomial time, then so can $\mathcal{P}_1$.

All hope is not lost when a problem is shown to be NP-complete. The set of NP-complete problems consist of many crucial real-world problems that need to be solved despite being hard. For example, many heuristics exist for various optimization problems that work well in practice, at least for certain subsets of the problem instances. Sometimes optimization problems admit approximation algorithms with guaranteed performance. See for example [7] for general discussions on approximation algorithms and complexity classes. In the majority of cases, however, the naive brute force algorithms do not work well in practice. Consider for example another famous NP-complete problem, namely vertex cover:

Vertex Cover
**Instance:** A graph $G$ and a non-negative integer $K \leq |V(G)|$
**Question:** Is there a set of vertices $V \subseteq V(G)$ of size $K$ such that $V$ covers $G$?

A set $V$ of vertices is said to cover $G$ if each edge of $G$ is incident to at least one vertex in $V$. The brute-force algorithm for this problem simply consists of checking every vertex subset of size $K$. There are $O(n^K)$ such subsets and as $n$ becomes large, checking them all, even for small $k$ is infeasible.

In 1986, Fellows and Langston [50], observed that vertex cover could be solved

in time $O(f(K)n^3)$. Later a very simple and elegant algorithm was discovered that runs in time $O(2^K n)$ [49]. Note how this time complexity separates the size of the input from the parameter $K$. The implication is that the algorithm is polynomial in the size of the input, and exponential only in the parameter. Hence, the problem is tractable even for large instances, as long as the minimal cover set is small. Improvements have since been made for vertex cover, and algorithms being able to handle $K$ up to about 400 have been implemented and used in multiple sequence alignment problems [27]. This is an example of parameterization of time complexity and the algorithm mentioned above for vertex cover is called a fixed-parameter tractable algorithm.

More formally, a paramterized problem is a subset of $\Sigma^* \times \mathbb{N}$. An instance of a paramterized problem is a pair $(I, K)$, where $K$ is the so-called parameter. The run-time of an algorithm for a parameterized problem is a function of $|I|$ and $K$. A parameterized problem is said to be fixed-parameter tractable (FPT) if there exists an algorithm for the problem with time complexity $O(f(K) \cdot |I|^c)$, where $c$ is a fixed constant and $f$ is a function of $K$ that does not depend on $I$. The parameterized version of vertex cover can be stated as follows:

$k$-Vertex Cover
**Instance:** A graph $G$ and a non-negative integer $K \leq |V(G)|$
**Parameter:** $K$
**Question:** Is there a set of vertices $V \subseteq V(G)$ of size $K$ such that $V$ covers $G$?

We note, in conclusion, that a problem may have many possible parameterizations. A problem can be fixed-parameter tractable for some parameterizations and not so for others. There is also a hierarchy of complexity classes in the theory of parameterized complexity. For a thorough treatment of this subject, we refer the interested reader to [43].

## 3.3 Maximum Likelihood Estimation with Expectation Maximization

Classic statistical inference can be divided into parametric and non-parametric. In non-parametric inference, no specific type of probability distribution or model is assumed. Instead, other kinds of hypotheses are made, for example, a common assumption is that the data are observations of independent and identically distributed (iid) random variables. It is, in general, quite difficult to incorporate previous knowledge or beliefs about the underlying real-world structure that has generated the data in non-parametric inference methods. In parametric inference, the observed data are assumed to be observations from some family of probability distributions. Examples of such distributions include the normal or Gaussian distribution, the multinomial distribution, and the Dirichlet distribution. Distributions may also be specified using probabilistic (generative) models containing structural elements attempting to capture the most important features of the real-

world situation that has generated the data. An example of such a distribution family is phylogenetic trees that we can think of as generating sequences. In any case, a distribution belonging to a family is determined by a set of parameters. For the normal distribution the parameters are the mean and variance, whereas for phylogenetic trees, parameters include the tree topology and the parameters of our chosen sequence evolution model (see Section 2.4). The classical inference problem is then to find, or estimate, the set of parameters that best fit the data according to some criteria.

The most widely used criteria for estimating the parameters is probably maximum likelihood (ML). The likelihood of a parameter set $\theta$ is simply the probability of the observed data $X$ given the parameters:

$$L(\theta|X) = p[X|\theta].$$

The maximum likelihood estimate of $\theta$ is defined as the $\theta^*$ maximizing the likelihood, or more generally, any function that is proportional to the likelihood. Note that the likelihood function is really a function of $\theta$ alone since we regard the data as fixed, and is not a probability distribution, i.e., in general $\int_\theta L(\theta|x) \neq 1$.

It is sometimes possible to determine the ML estimate by deriving closed formulas, but in many cases such a method is infeasible. A popular computational technique for parameter estimation is Expectation Maximization (EM). The method has been in use in different forms for a long time, but was generalized and popularized with the publication of a paper by Dempster, Laird, and Rubin in 1977 [37]. For some notes on the history of the EM algorithm, see [109].

EM has been successfully applied to a wide variety of problems in diverse scientific fields and many modifications and improvements have been suggested, see for example [109, 79, 110, 55]. In the next subsection, we will give the standard derivation of the EM algorithm together with a proof of its convergence. Subsequently, we will discuss the structural EM algorithm of Friedman *et al.* [60, 61] which is directly related to the work presented in Paper IV of this thesis.

### Standard EM

Let $X = \{x_1, \ldots, x_N\}$ be the set of observed data, and let $\theta$ denote the set of parameters. In many applications, we also have a set of missing data or hidden variables. These are sometimes introduced in the model in order to simplify computations of certain probabilities. As a concrete example, assume that $X$ consists of a set of points on the real line and that we wish to model the data using a mixture $Y$ of two normal distributions:

$$\begin{aligned} Y_1 &\sim N(\mu_1, \sigma_1^2), \\ Y_2 &\sim N(\mu_2, \sigma_2^2), \\ Y &= \begin{cases} Y_1 & \text{with probability } \pi, \\ Y_2 & \text{with probability } 1 - \pi. \end{cases} \end{aligned}$$

Thinking of the model as generative, the above notation has the following interpretation: each data point is generated from distribution $Y_1$ with probability $\pi$ or $Y_2$ with probability $1-\pi$. In this case, $\theta$ consists of five parameters:

$$\theta = (\pi, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2).$$

The computation of certain probabilities are made easier by introducing a set $Z$ of hidden variables indicating the distribution from which each data point in $X$ was generated:

$$z_i = \begin{cases} 0 & \text{if } x_i \text{ was generated from distribution } Y_i, \\ 1 & \text{otherwise.} \end{cases}$$

The idea of the EM algorithm is as follows. At the start of the $n$th iteration we have a set of parameters $\theta_n$. Together with the observed data $X$, $\theta_n$ induces a probability distribution on the hidden variables, $\Pr[Z|X,\theta_n]$. The set of parameters $\theta_{n+1}$ for the next iteration is obtained by finding the $\theta$ that maximizes the expectation of the so-called complete-data log-likelihood

$$E_{Z|X,\theta_n}\big[\log p[X,Z|\theta]\big],$$

where the expectation is taken over the distribution $\Pr[Z|X,\theta_n]$. The procedure is guaranteed not to decrease the likelihood, in other words $L(\theta_{n+1}|X) \geq L(\theta_n|X)$. We next show why this actually works.

First, we show that the likelihood can be written as the sum of two expectations:

$$\begin{aligned}
\log L(\theta|X) &= \log p[X|\theta] \\
&= \log p[X|\theta] \cdot 1 \\
&= \log p[X|\theta] \left( \sum_Z \Pr[Z|X,\theta] \right) \\
&= \sum_Z \Pr[Z|X,\theta] \log p[X|\theta] \\
&= \sum_Z \Pr[Z|X,\theta] \log \frac{p[X,Z|\theta]}{\Pr[Z|X,\theta]} \\
&= \sum_Z \Pr[Z|X,\theta] \log p[X,Z|\theta] - \sum_Z \Pr[Z|X,\theta] \log \Pr[Z|X,\theta] \\
&= E_{Z|X,\theta}\big[\log p[X,Z|\theta]\big] - E_{Z|X,\theta}\big[\log \Pr[X|Z,\theta]\big]. \qquad (3.1)
\end{aligned}$$

For the next result, we use Jensen's inequality to obtain

$$\begin{aligned}
\log L(\theta|X) &= \log p[X|\theta] \\
&= \log \sum_Z p[X, Z|\theta] \\
&= \log \sum_Z \Pr[Z|X, \theta'] \frac{p[X, Z|\theta]}{\Pr[Z|X, \theta']} \\
&= \log E_{Z|X,\theta'} \left[ \frac{p[X, Z|\theta]}{\Pr[Z|X, \theta']} \right] \\
&\geq E_{Z|X,\theta'} \log \left[ \frac{p[X, Z|\theta]}{\Pr[Z|X, \theta']} \right] \qquad \text{(By Jensen's inequality)} \\
&= E_{Z|X,\theta'} \big[ \log p[X, Z|\theta] \big] - E_{Z|X,\theta'} \big[ \log \Pr[X|Z, \theta'] \big]. \qquad (3.2)
\end{aligned}$$

Noting the similarity between (3.1) and (3.2), we define the famous $Q$- and $R$-terms as

$$\begin{aligned}
Q(\theta, \theta') &= E_{Z|X,\theta'} \big[ \log p[X, Z|\theta] \big] \\
R(\theta) &= -E_{Z|X,\theta} \big[ \log \Pr[X|Z, \theta] \big]
\end{aligned}$$

Assume that we have a set of parameters $\theta_n$ at the start of the $n$th iteration and let

$$\theta^* = \operatorname*{argmax}_\theta Q(\theta, \theta_n). \qquad (3.3)$$

We now have that

$$\begin{aligned}
\log L(X|\theta^*) &\geq Q(\theta^*, \theta_n) + R(\theta_n) && \text{(By (3.2))} \\
&\geq Q(\theta_n, \theta_n) + R(\theta_n) && \text{(By definition of } \theta^*) \\
&= L(X|\theta_n). && \text{(By (3.1))}
\end{aligned}$$

Hence, by choosing $\theta_{n+1} = \theta^*$, we are guaranteed not to decrease the likelihood. All we need to do to implement an EM algorithm is to perform the maximization of the Q-term in (3.3). The maximization of the Q-term is generally done in two steps: In the E-step, certain quantities are computed that only depend on the current set of parameters and the observed data. This is in preparation for the M-step where the computed quantities are used to find the set of parameters that maximizes the Q-term.

We note here that maximization of the Q-term is not necessary for convergence. Convergence to a local optimum is guaranteed as long as we are able to find a set of parameters $\theta_{n+1}$ such that $Q(\theta_{n+1}, \theta_n) \geq Q(\theta_n, \theta_n)$. This procedure is called Generalized EM (GEM) and can be used when maximization of the Q-term is infeasible. One drawback of GEM compared to standard EM is a potentially slower rate of convergence.

In conclusion, both standard EM and generalized EM may suffer from the same drawbacks as local search methods, and finding a globally optimal solution may require different heuristics such as using a set of random start values or simulated annealing strategies.

### Structural EM

In 1997, Friedman devised a structural EM algorithm for Bayesian networks, that beside improving the numeric parameters in each step, also improves the structure [60]. In 2002, the same approach was used for tree reconstruction [61].

Likelihood-based methods for tree reconstruction have been very successful and are quite popular. Prior to Friedman's contribution, methods for ML estimation of phylogenies used the EM algorithm only for optimization of the parameters on a fixed tree. When searching for the best topology, each tree considered would have to be passed to the EM algorithm for parameter optimization. This is computationally very expensive, and in practice, only a few selected topologies could be considered. Friedman *et al.* observed that it is possible to simultaneously improve the topology and parameters.

In this setting, the input consists of aligned sequences $X$, and the set of parameters of the model consist of both the tree topology $T$ and the lengths $l$ of the tree edges, i.e., $\theta = (T, l)$. Just as in standard EM, the parameters $\theta_n$ of the previous iteration induce a distribution on the space of topologies and lengths. The crucial observation made by Friedman *et al.* is that the contribution to the Q-term from each *possible* edge is the same for all trees and can be computed once and for all. These contributions are then used as weights on the set of all pairs of vertices and the problem of finding the best topology given $\theta_n$ is reduced to finding the bifurcating tree with greatest total weight. Unfortunately, this problem turns out to be NP-complete. To overcome this difficulty, Friedman *et al.* use the maximum spanning tree algorithm to obtain a tree that is not necessarily bifurcating, but which is then transformed into a bifurcating tree via modifying steps that are guaranteed not to decrease the likelihood. Hence, in each iteration of the EM algorithm, both the topology and the parameters are changed, leading to great savings in computational time.

We note here that an important distinction can be made among structural EM algorithms. Just as for standard EM and generalized EM, we can distinguish between structural EM algorithms that respectively, maximize and merely improve on the Q-term. When possible, an EM algorithm that maximizes the Q-term in each iteration is preferred due to faster convergence rates. In Paper IV of this thesis, we provide a structural EM algorithm that maximizes the Q-term in each step. In order to emphasize the distinction between maximizing and improving the Q-term, we call our algorithm a *global* structural EM algorithm.

## 3.4 Bayesian Inference with Markov Chain Monte Carlo

Bayesian statistical inference has become increasingly popular in the field of bioinformatics. In this section, we will give a brief background on Bayesian statistics and touch on some of the controversial issues. Finally, we will provide a discussion on MCMC techniques that is relevant to the work presented in Paper III of this thesis.

### A Philosophical Question

Discussions of Bayesian versus classical statistics usually start with a philosophical question: what is a probability? The so-called frequentist answer is that the probability of an event $A$ is the long run proportion of times that event $A$ occurs during a large number of replications of an experiment. Hence, the probability of heads in a coin tossing experiment with a fair coin is 0.5 and the probability of obtaining a six on a roll of a single fair die is $\frac{1}{6}$. In contrast, the Bayesian answer is that probability is a measure of an individual's uncertainty about the outcome of an experiment, with the added constraint that the individual's opinion must be consistent, in other words, assignment of probabilities to events must be in accordance with the Kolmogorov axioms of probability theory. In the case of a toss of a coin and a roll of a die above, most Bayesians too would assign probabilities 0.5 and $\frac{1}{6}$ to the events of heads and six, respectively. However, to a Bayesian, *any* event can be assigned a probability as a measure of uncertainty, even if the experiment could never be replicated. More importantly, a Bayesian may assign a probability to a hypothesis, while to a frequentist, a hypothesis should either be rejected or retained. A highly cited and enjoyable classic paper on the subject of Bayesian subjective probability was written by Edwards, Lindman, and Savage [46].

Taking an example from [46], imagine a great prize being offered to predict the outcome of a coin toss. With no other previous knowledge, both the frequentist and the Bayesian statistician would assign a probability of 0.5 to the event of heads. Assume now that once a prediction has been made, you are informed that the coin has either two heads or two tails. This is a point of departure between the frequentist and the Bayesian. While the Bayesian, having no other knowledge, is likely to assign a probability of 0.5 to the hypothesis that the coin has two heads, to the frequentist it would seem that no such probability can be assigned. In any case, it would be hard to see why either the Bayesian or the frequentist would have any reason to change their predictions.

### Bayes's Theorem, Priors, and Posteriors

The name "Bayesian" comes from the frequent application of Bayes's theorem in Bayesian inference. Bayes's theorem simply relates the marginal and conditional

probabilities of two events:

$$\Pr[A|B] = \frac{\Pr[B|A]\Pr[A]}{\Pr[B]},$$

when $\Pr[A] > 0$ and $\Pr[B] > 0$. The theorem itself is in no way controversial and is valid under both the frequentist and Bayesian interpretation of probability. The controversy arises with the use of priors and computation of posterior probabilities. Assume that we have a data set $D$ which we believe to be generated by one of $n$ different fully specified models $M_1, \ldots, M_n$, and that the conditional probabilities $\Pr[D|M_i]$ are well-defined and easily computable. To choose among the models, we could for example use maximum likelihood and pick the model for which $\Pr[D|M_i]$ is greatest. The Bayesian approach is instead to compute posterior probabilities on the set of models using Bayes's theorem:

$$\Pr[M_i|D] = \frac{\Pr[D|M_i]\Pr[M_i]}{\Pr[D]}.$$

Using the law of total probability, the denominator of the right hand side of the above equation can be written as

$$\Pr[D] = \sum_{i=1}^{n} \Pr[D, M_i] = \sum_{i=1}^{n} \Pr[D|M_i]\Pr[M_i].$$

The probability $\Pr[M_i]$ is called a prior and represents our belief that model $M_i$ is the correct model *before* observing the data. The probability $\Pr[M_i|D]$ is called the posterior and represents our updated belief that $M_i$ is the correct model *after* having observed the data.

An example of a continuous case is when we want to estimate a parameter of a model. Assume that we know that a distribution $M$ has generated the data, but the parameter $\theta$ of $M$ is unknown. The distribution $p[\theta]$ represents how likely different values of $\theta$ are prior to having seen the data. After data has been gathered, the posterior distribution of $\theta$ is given by Bayes's theorem:

$$p[\theta|D] = \frac{\Pr[D|\theta]p[\theta]}{\int \Pr[D|\theta]\Pr[\theta]d\theta}.$$

We can then estimate $\theta$ using the posterior mean $E[\theta|D] = \int \theta p[\theta|D]d\theta$, or we can find an interval $(a, b)$ such that, for given $\alpha$, $\Pr[a < \theta < b|D] = \int_a^b p[\theta|D]d\theta = 1-\alpha$.

The use of priors continues to be controversial, and a thorough discussion lies outside the scope of this thesis. One objection to the use of priors is that they are not objective, but see [14] for a perspective on objectivity in Bayesian and classic statistics. Another is that it may be difficult to express one's beliefs in terms of a probability distribution and instead one might choose a certain prior for practical reasons rather than to express one's true opinions. And besides, humans often harbor inconsistent opinions that contradict the axioms of probability theory.

The good news is however that priors do not always have a strong influence on the results. It can be shown that under certain conditions, an increasing amount of data will lead to more and more similar posterior distributions [16]. Others have argued for the use of non-informative priors as being an objective choice when there is a lack of prior opinion, see for example [81].

There are often other benefits to Bayesian analysis compared to classical statistics that compensate for the use of priors. For example, in phylogenetics, MCMC techniques have enabled the use of arbitrary prior distributions, and a more efficient investigation of the state space. Bayesian methods allow us to efficiently deal with high dimensional models, and to obtain marginal distributions on the parameters of interest. Also, Bayesian methods allow us to explore posterior distributions rather than just summary statistics such as mean and variance. For an introduction to Bayesian statistics, see the excellent book by Peter Lee [95].

### MCMC

In Bayesian inference, we are often confronted with various integration and optimization problems. We frequently need to find a marginal distribution or compute expectations. When dealing with large dimensional spaces, analytic solutions are usually not readily available. Instead, we can obtain Monte Carlo estimates by drawing iid samples from a target distribution. These samples can then be used to approximate the density or expectation of interest. Assume for example that we draw a set of iid samples $x_1, \ldots, x_N$ from a density $p[x]$ defined on a high-dimensional space $X$. The integral

$$\int_X f(x)p[x]dx$$

can then be approximated by the sum

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i).$$

Markov Chain Monte Carlo is a method based on Markov chains that allows us to obtain samples from non-standard distributions from which we cannot draw samples directly. To use MCMC techniques we need to be able to evaluate at least ratios of the target distribution.

First, we describe MCMC on finite state spaces. Assume that the state space is $S = \{s_1, \ldots, s_n\}$. A Markov chain on $S$ is a sequence of random variables $X_1, X_2, \ldots$ taking values in the state space $S$ such that

$$\Pr[X_n|X_{n-1}, \ldots, X_1] = \Pr[X_n|X_{n-1}].$$

The chain is called *homogeneous* if

$$\Pr[X_n = s|X_{n-1} = s'] = \Pr[X_j = s|X_{j-1} = s'],$$

for all $j, n > 1$. In other words, a homogeneous Markov chain lacks memory and the distribution on the states for the next step is fully determined by the current state. A Markov chain on a finite state space can be represented by a transition matrix $T$, where $T_{ij} = \Pr[X_n = s_j | X_{n-1} = s_i]$ and $\sum_j T_{ij} = 1$. Now, assume that we pick a start state randomly from a distribution represented by a row vector $\pi = (\Pr[s_1], \ldots, \Pr[s_n])$. The distribution on the states after one transition is given by $\pi T$. In general, the probability distribution on the states after $n$ transitions is given by $\pi T^n$. If the transition matrix is both irreducible and aperiodic, then the distribution on the states will converge to a unique invariant distribution $\mu$ irrespective of the start distribution $\pi$, i.e.,

$$\pi T^n \to \mu \qquad \text{as } n \to \infty,$$

for any distribution $\pi$. A transition matrix is irreducible if any state is reachable from any other in a finite number of steps. A state $i$ has period $k$ if the length of any path returning to $i$ is always a multiple of $k$. A transition matrix with a state of period $k > 1$ is called periodic. Otherwise, it is called aperiodic.

A sufficient condition to ensure that $\mu$ is the desired distribution $p$ from which we want to sample is the so-called *detailed balance* criterion:

$$p_i T_{ij} = p_j T_{ji}.$$

When the state space is infinite, we can instead use the Metropolis-Hastings algorithm [111, 73]. Assume that we want to sample from a distribution on a multi-dimensional state space $\Theta = (\Theta_1, \ldots, \Theta_m)$. The components of $\Theta$ can be either discrete, continuous, or a mix of discrete and continuous variables. More generally, each component can be a multi-dimensional random variable in itself. As an example, consider phylogeny where $\Theta$ could be a phylogenetic tree together with lengths associated with the edges.

Instead of a transition matrix, the Markov chain is now specified as a set of proposal distributions, $R_k(\Theta'_k | \Theta)$, $k = 1, \ldots, m$, that given the current state $\Theta$ propose a change in one of the components $\Theta_k$ according to some distribution. In other words, the proposed state $\Theta'$ obtained from $R_k$ differs from $\Theta$ only in the $k$th component: $\Theta_i = \Theta'_i$, $i \neq k$. The proposed new state is then accepted with probability $A(\Theta, \Theta')$, in which case $\Theta'$ becomes the new current state, or is rejected and the current state remains $\Theta$. The component to change in each step is often picked randomly in each step. In the original Metropolis algorithm, the acceptance probability is

$$A(\Theta, \Theta') = \min\left(1, \frac{\Pr[\Theta']}{\Pr[\Theta]}\right),$$

so that if the probability of the proposed state is greater than the current state, it is always accepted, and otherwise, it is accepted with probability $\Pr[\Theta'] / \Pr[\Theta]$. One way to achieve detailed balance under this acceptance strategy is to ensure that

$$R_k(\Theta'_k | \Theta) = R_k(\Theta_k | \Theta').$$

Hastings generalized the Metropolis algorithm to allow non-symmetric proposal distributions. The acceptance probability is then

$$A(\Theta, \Theta') = \min\left(1, \frac{\Pr[\Theta']R_k(\Theta_k|\Theta')}{\Pr[\Theta]R_k(\Theta'_k|\Theta)}\right).$$

This again ensures detailed balance.

For more on Metropolis algorithms and MCMC, and discussions on convergence rates, tests of convergence, and more, see for example [120, 98].

# Chapter 4

# Computational Methods and Models for Duplications and LGTs

This chapter gives an overview of the different phylogenetic methods concerned with gene duplications and LGTs. Algorithms have been developed for a variety of problems, such as tree reconciliation, species tree reconstruction, and orthology analysis. Tree reconciliation is the problem of explaining the differences between a species tree and a corresponding gene tree by giving a plausible evolutionary history of the latter inside the former. Species tree reconstruction refers to problems in which an optimal species tree is sought when given a set of incongruent gene trees. Alternatively, the input can consist of sequences from different gene families, in which case substitution models are also taken into account when seeking optimal species trees. In orthology analysis, the problem is determining whether or not pairs of sequences are orthologous. Data in this case can consist of either trees, sequences, or a mix of trees and sequences.

The first section of this chapter deals with the observation that certain problems in parasitology and biogeography are analogous to those of molecular evolution. The subsequent three sections give a background on previous work on the problems mentioned in the previous paragraph. A common feature of all the methods discussed is their focus on duplications and LGTs. Sections 4.5 and 4.6 discuss the work presented in Papers I, II, and III of this thesis.

## 4.1 Trees Within Trees

The notion of a tree structure evolving inside another tree structure has been used in at least three separate disciplines: molecular evolution, parasitology, and biogeography. In each case, questions arise about how a *host* is tracked by an *associate*. In molecular evolution, genes track organisms; in parasitology, parasites track hosts; and in biogeography, organisms track areas. The structure most widely used to depict histories of hosts and associates is that of a tree. Different historic events

cause the trees of hosts and their associates to be incongruent, thus creating the need to specify exactly what those events are and where they have occurred. As it turns out, each event considered in one discipline has an analogue in the others and results in the same type of incongruity between host and associate trees. The fundamental similarity between the problems in the different disciplines was not recognized until the 1990s, although similar work in the different disciplines had been done independently. A clear exposition on this subject can be found in [129].

In molecular evolution, the events under consideration are speciations, duplications, LGTs, and losses. The corresponding events in parasitology are co-speciation, independent parasite speciation, host switching, and lineage sorting, respectively. In biogeography we have vicariance, sympatry, dispersal, and extinction. The fundamental observation here is that a single model in which an associate tree evolves inside a corresponding host tree is adequate to capture all three cases. In fact, we can find more examples where such a model can be applied; for example, the evolution of protein domains inside gene trees.

In the following sections, we will formulate problems and discuss methods using terms from molecular evolution.

## 4.2 The Duplication-Loss Model

In the duplication-loss model, we assume that any incongruities between an organismal tree and a corresponding gene tree are due to duplications and losses. In other words, we assume that the mode of genetic transfer is strictly vertical from parent to child, although genes can be duplicated or lost and this change in genetic make-up is sometimes spread to the entire population and is fixed. Clearly, the history of a set of homologous genes represented by a gene tree is then restricted to having occurred inside the edges of a corresponding species tree. Each internal gene tree vertex corresponds to either a duplication event or a speciation event. See Figure 4.1 where a gene tree is drawn inside a species tree showing the evolutionary history of a set of genes; this is an example of tree reconciliation in which a biologically feasible explanation is provided for the disparity between the host and associate trees.

In 1970, Fitch [57] made a distinction between paralogous and orthologous genes, i.e., genes whose least common ancestor in the gene tree is a duplication or speciation, respectively. Similar concepts had been developed much earlier in parasitology, see for example [28]. The development of methods for detecting pairs or groups of orthologous genes is an important step in the prediction of gene function. Traditionally, trees are taken as the data to be analyzed. A species tree is assumed given, together with a gene tree that has been constructed from sequences using methods analogous to those discussed in Section 2.4. Given a species tree and a corresponding gene tree, an important problem is determining the evolutionary history of the gene tree within the species tree and to answer questions such as *which pairs of genes are orthologous and which paralogous?*.

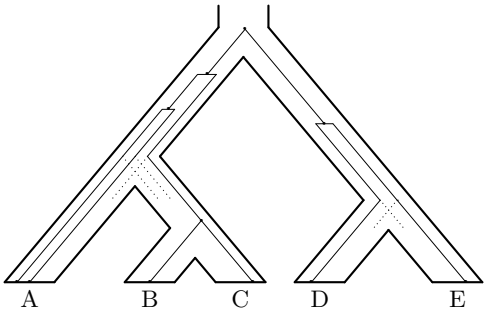



Figure 4.1: ***Example of genes evolving inside a species tree according to the duplication-loss model.***

We will use the following notation in the discussions below. We take all trees to be rooted binary trees. The edges of a rooted tree are assumed to be directed away from the root. The subtree of a tree $T$ rooted at the vertex $u$ is denoted $T_u$. The correspondence between a species tree $S$ and a gene tree $G$ is given via a leaf-mapping function $\sigma : L(G) \to L(S)$ that maps each gene to the extant species to which it belongs. For convenience, we assume that $\sigma$ is extended to map sets of gene tree leaves to the corresponding sets of species tree leaves. The function mapping a set of tree vertices to their least common ancestor is denoted lca .

In 1979, Goodman *et al.* gave a parsimony method for tree reconciliation that maps the gene tree inside the species tree such that the number of inferred duplications and losses is minimized. This mapping is called the least common ancestor mapping, $\lambda : V(G) \to V(S)$, and is defined by

$$\lambda(u) = \text{lca}\,(\sigma(L(G_u))).$$

Note that $\lambda$ maps vertices of $G$ to vertices of $S$. When describing a possible evolutionary history $G$ inside $S$, the gene tree vertices representing speciation events are associated with the species tree vertex that corresponds to the same speciation event. A duplication vertex in $G$ is associated with the edge of $S$ in which the duplication occurred. The interpretation of the mapping given by $\lambda$ is that if $u \in V(G)$ represents a speciation, then $\lambda(u)$ represents that same speciation event, and if $u$ is a duplication, then the duplication occurred along the incoming edge of $\lambda(u)$.

Building on the framework provided by Goodman *et al.*, Guigó *et al.* attempted to find the species tree whose reconciliation with a set of gene trees requires a minimum number of duplications [65]. Their method can be described as a heuristic local search method or hill-climbing where the neighborhood of a tree is defined by nearest neighbor interchange (NNI) operations [116]. Ma *et al.* proved hardness results for several species tree reconstruction methods [102], and several heuristic methods for species tree reconstruction have been developed [136, 128, 151].

Assuming that at least one gene tree has had a constant number of lineages in each species tree lineage, there is an FPT algorithm for reconstructing the optimal species tree [68].

Going beyond parsimony methods, probabilistic models of gene evolution for the duplication-loss model have recently been proposed. In [4, 5, 139, 155], a complete framework for computational analysis in a probabilistic setting has been developed. The model is most conveniently described as a generative model that generates a gene tree and sequences on a give species tree with times associated with the species tree edges. The model of evolution is based on the standard birth-death process [86] which generates duplications and losses along the edges of a species tree resulting in a gene tree. Sequences are then generated according to an arbitrary choice of standard substitution models. Adopting a Bayesian approach and using MCMC techniques, it is possible to compute various posterior probabilities of interest such as the probability of a gene tree given sequences, or the probability of two sequences being orthologous or paralogous. Posterior probability distributions of duplication and loss rates can also be studied. The latest development in this direction extends the model with an iid model of sequence evolution rate variation across gene tree edges [155]. Methods that simultaneously consider sequence evolution and gene tree and species tree reconciliation when identifying duplications have been termed duplication analysis. An *ad hoc* method for duplication analysis that also takes gene order information into account was presented in [149].

## 4.3 The Transfer-Loss Model

Early attempts at defining evolutionary models taking LGTs into account include the network model [147, 74, 75]. A related approach considers the subtree transfer operation on a tree in which a subtree is moved to a different location. The corresponding optimization problem is to find a minimal set of subtree transfer operations that transform one given tree to another [34, 33, 76]. Nakhleh *et al.* developed a heuristic for phylogenetic network reconstruction given a species tree and a set of gene trees [119].

Analogous to the case of duplications, tree reconciliation problems have been considered in settings where only transfers and losses are take into account. Variations on parsimony problems were defined and considered in [69].

Probabilistic models have also been suggested. In [82], a model was described in which sequences evolve along a network. Huelsenbeck *et al.* developed a Bayesian framework in the context of hosts and parasites for detecting host switches in which the data to be analyzed consists of host and parasite sequences. The model considers only the case where each host is tracked by a single parasite species so that when a host acquires a new parasite, the parasite formerly associated with the host becomes extinct. In [97], a generative model for LGT, without duplications and losses, based on a Poisson process rather than a birth-death process, was presented and used to generate synthetic data. Biological data was also compared with synthetic data

in order to infer LGT rates. In [17], Boc and Makarenkov develop a method for detecting LGTs based on distances between the sequences used to infer the species tree and gene tree. A heuristic algorithm for inferring a network using a minimum number of LGTs from a species tree and a corresponding set of gene trees was developed by Nakhleh *et al.* in [119].

Other, non-phylogenetic, methods for detection of LGTs include the use of atypical sequence composition, which can be used to detect recent LGTs, see for example [9].

## 4.4 The Duplication-Transfer-Loss Model

Based on the idea of reconciled trees, Charleston developed a computer program, Jungles [26], attempting to solve a parsimony variant of tree reconciliation that considers both duplications and transfers. Unfortunately, the presentation lacks mathematical rigor and is plagued by errors in proofs. The time complexity of the method was not analyzed, and in fact, is probably exponential. Further evidence to support this conjecture is found in Paper I of this thesis where it is shown that finding most parsimonious reconciliations that are temporally feasible is NP-hard.

Although probabilistic models of gene evolution for the duplication-transfer-loss model (DTL-model) have been defined, see for example [32], these have not previously been used to infer transfers or to reconcile trees. A probabilistic model based on the birth-death process, as described in Paper III of this thesis, was in fact used to produce synthetic data for analysis in [67]. In [32], a similar model was suggested, but was applied only to gene family sizes. The two models differ in that the model in [32] assumes that the transfer rate is constant and independent of the number of gene lineages currently present.

## 4.5 DTL-scenarios

In this section, we discuss the combinatorial model and methods for tree reconciliation presented in Papers I and II. The work is a contribution along the lines of the work of Goodman *et al.* but for the much more complicated case when both duplications, transfers, and losses are considered.

We consider as input a species tree $S$ and a gene tree $G$, both rooted binary trees. The association of genes with species is given by a leaf-mapping function $\sigma : L(G) \to L(S)$. For convenience, we extend $\sigma$ to map sets of gene tree leaves to the corresponding sets of species.

Given $S$, $G$, and $\sigma$, our aim in Paper I is to find the most parsimonious reconciliation explaining the evolution of $G$ with respect to $S$. Adding LGT as an evolutionary event yields a complexity that requires strict formal definitions. In order to achieve both biological and mathematical soundness in our definitions, we introduce the concept of DTL-scenarios whose associated costs are defined as the number of duplications and LGTs. We do not need to consider every biologically

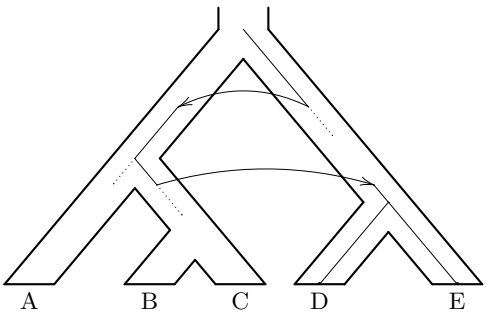



Figure 4.2: ***Example of a non-parsimonious reconciliation in the DTL-model.***

possible reconciliation between $S$ and $G$ in a parsimony setting, and therefore, the definition of DTL-scenarios have been carefully crafted to ensure both biological feasibility as well as non-redundancy. Consider, for example, the evolutionary history shown in Figure 4.2. Although the example is biologically possible, there is no need to consider such histories in a parsimony setting. A thorough justification for our definition is given in Paper II.

A DTL-scenario for $S$, $G$, and $\sigma$ consists of a partition $\{\Sigma, \Delta, \Theta\}$ of the internal vertices of $G$, a subset $\Xi \subset E(G)$, and a function $\gamma : V(G) \to V(S)$. The subset $\Xi$ consists of all the transfer edges of $G$. The parts $\Sigma$, $\Delta$, and $\Theta$ correspond to the speciation, duplication, and transfer vertices of $G$, respectively. Finally, $\gamma$ maps the gene tree into the species tree indicating where the speciations, duplications, and lateral gene transfers have occurred. Formally, A DTL-scenario for a species tree $S$, a gene tree $G$, and a leaf-mapping function $\sigma : L(G) \to L(S)$ is an octuple

$$(S, G, \sigma, \gamma, \Sigma, \Delta, \Theta, \Xi),$$

where $S$ and $G$ are rooted binary trees, $\sigma : L(G) \to L(S)$ is a leaf-mapping function, $\gamma : V(G) \to V(S)$ is an extension of $\sigma$, $\Sigma$, $\Delta$, and $\Theta$ form a partition of the internal vertices of $G$, and $\Xi \subset E(G)$ is a subset of the gene tree edges such that:

(I) If $u$ is an internal gene tree vertex with children $v$ and $w$, then
   a) $\gamma(u)$ is not a proper descendant of $\gamma(v)$ or $\gamma(w)$
   b) At least one of $\gamma(v)$ and $\gamma(w)$ is a descendant of $\gamma(u)$

(II) $(u, v) \in \Xi$ if and only if $\gamma(u)$ is incomparable to $\gamma(v)$

(III) If $u$ is an internal gene tree vertex with children $v$ and $w$, then
   a) $u \in \Theta$ if and only if $(u, v) \in \Xi$ or $(u, w) \in \Xi$
   b) $u \in \Sigma$ only if $\gamma(u) = \text{lca}\,\{\gamma(v), \gamma(w)\}$ and $\gamma(v)$ and $\gamma(w)$ are incomparable

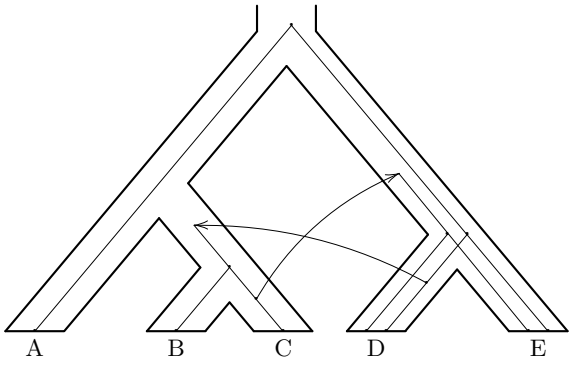



Figure 4.3: ***An example of a cyclic DTL-scenario.***

c) $u \in \Delta$ only if $\gamma(u)$ is an ancestor of lca $\{\gamma(v), \gamma(w)\}$

We also need to consider the fact that some sets of LGTs can lead to temporally infeasible reconciliations, an example of which is shown in Figure 4.3. We say that a DTL-scenario is acyclic if and only if

(V) There is a total order $<$ on $V(S)$ such that

a) if $(x, y) \in E(S)$, then $x < y$

b) if $(u, v), (u', v') \in \Xi$ and $v$ is an ancestor of $v'$, then $p(\gamma(u)) < \gamma(v')$

In Paper II, we prove that the condition of acyclicity given above is both sufficient and necessary to ensure temporal feasibility.

A major result in Paper I is that finding most parsimonious acyclic DTL-scenarios is NP-hard. However, earlier results suggest that in most data-sets cyclicity is usually not a problem. Therefore, dropping the requirement of acyclicity we develop a polynomial-time dynamic programming (DP) algorithm as well as an FPT-algorithm for finding most parsimonious DTL-scenarios. Our algorithms are applied to biological data that have been previously analyzed in the literature with respect to LGTs.

In Paper II, we extend our model by allowing arbitrary costs to be associated with duplications and LGTs and give a DP algorithm for finding minimal-cost DTL-scenarios. The algorithm in Paper II constitutes a considerable improvement on the time complexity of the DP algorithm in Paper I.

For any species tree and gene tree pair, there are only a finite number of DTL-scenarios. The algorithms mentioned so far are able to find optimal DTL-scenarios for any given cost scheme. Due to the combinatorial nature of the problem, there are sets of cost schemes with the same set of optimal DTL-scenarios. An interesting computational problem is to partition the space of cost schemes based on the sets of optimal DTL-scenarios. This is analogous to the problem of parametric sequence

alignment [66]. In Paper II, we give a polynomial-time algorithm for parametric tree reconciliation. With this algorithm at our disposal, we are able to obtain the set of all DTL-scenarios that are optimal under *any* cost scheme. We then use this method to perform tests on synthetic data, yielding very encouraging results, that show the trade-off between sensitivity and specificity for different cost schemes.

## 4.6 A Comprehensive Probabilistic Model of Gene Evolution

In Paper III, we develop a probabilistic model of gene evolution with duplications, LGTs, and losses. To our knowledge, this is the first such probabilistic model that has been used for inference of duplications and LGTs.

We assume that a fixed species tree is given with divergence times associated with its vertices. The model is best described as first generating a gene tree with branch lengths after which some standard substitution model can be used to generate sequences. The model uses a standard birth-death process to generate a gene tree with respect to the species tree given rates for duplications, LGTs, and losses. The resulting gene tree has times associated with its edges. We achieve a relaxed molecular clock by assuming that substitution rates on gene tree edges are iid $\Gamma$-distributed variables. The rates obtained from the $\Gamma$-distribution, together with edge times, induce branch lengths on the edges of the gene tree. Finally, a substitution model is used to generate sequences. In Paper III, we use the JTT model, but any standard substitution model can be used.

Many interesting computational problems can be defined based on the model described above. We provide a Bayesian framework for the analysis of sequence data using MCMC techniques. We use priors on the parameters of the model, namely the rates of duplication, LGT, and gene loss and the mean and variance of the $\Gamma$ distribution. A state in our Markov chain is a triple $(G, l, \theta)$, where $G$ is a gene tree, $l$ is a function assigning branch lengths to the edges of $G$, and $\theta$ is the set of birth-death rates and the mean and variance of the $\Gamma$ distribution. By using standard MCMC techniques, interesting posterior distributions can be studied. Examples include the posterior distribution on the gene tree topologies, the LGT or duplication rate, and the number of LGTs.

In order to use MCMC, we need to be able to compute ratios of posterior probabilities of the form $\Pr[G, l, D|\theta]$, where $D$ is the data to be analyzed in the form of sequences (since the species tree is fixed, we omit it from our notation). We can rewrite the probability of a state in our Markov chain as

$$\Pr[G, l, \theta|D] = \frac{\Pr[D|G, l] \Pr[G, l|\theta] \Pr[\theta]}{\Pr[D]}.$$

When computing ratios of posterior probabilities, the denominator in the above expression will cancel, and therefore, we do not need to compute $\Pr[D]$. $\Pr[\theta]$ is simply our prior distribution on the parameters, and $\Pr[D|G, l]$ can be computed

according to our chosen substitution model. A major contribution of Paper III is an algorithm for computing the probability $\Pr[G, l|\theta]$. More specifically, we approximate $\Pr[G, l|\theta]$ by introducing discretization points on the species tree and applying a mix of dynamic programming algorithms and techniques from numerical analysis.

# Chapter 5

# Modeling Cancer Progression

This chapter provides a short background on cancer progression models, and Section 5.2 contains a description of the work presented in Paper IV.

## 5.1 Overview of Current Methods

Mathematical modeling of cancer progression started more than fifty years ago with simple, yet groundbreaking, models of tumorigenesis [122, 3, 90]. The early models all assumed that cancer is a stochastic multistep process with small transition rates. A more recent example in that direction is [83]. As noted in Section 2.6, cancer progression is an evolutionary process, and therefore, it is not surprising that methods and models from population genetics have been used extensively, see [113] for a review.

In this thesis, we will follow a different line of research, which started with the introduction of Oncogenetic Trees (OTs) by Desper *et al.* [38]. Since Vogelstein's path model of colon cancer, numerous narrative models for progression of diverse cancer types have been suggested, for example [80, 141, 146]. Such models are often the result of *ad hoc* handmade reconstructions. The introduction of OTs was an attempt at a more stringent mathematical modeling of cancer progression. An OT is a rooted tree where each vertex represents a specific genetic aberration and there is a probability associated with each edge. An OT generates a set of aberrations by first choosing a set of edges, each independently and according to its associated probability. The set of vertices reachable from the root, using only the chosen edges, is then the set of generated aberrations. In this way, an OT induces a probability distribution on the power set of all aberrations.

Given cross-sectional data, i.e., sets of aberrations where each set is from a unique tumor or patient, the computational task is to find the correct OT. Desper *et al.* showed that computing a specific weight function on the set of all pairs of vertices and then using Edmonds's maximum branching algorithm to obtain the topology, the correct tree will be recovered with high probability.

One problem in OTs is that once progression stops at some vertex $u$, i.e., when none of the outgoing edges of $u$ are chosen in the first step, then progression cannot reach any of the descendants of $u$. Biological data is almost always noisy, and in any case, real cancer progression is not tree like. The result is that usually every OT, except the OTs with a star topology, assign zero probability to some of the data, and therefore, using likelihood methods is not straightforward. Another problem is that cancer progression is best described using acyclic graphs that allow an aberration to be obtained via different pathways.

In an attempt to capture more of the graph-like progression of cancers, Beerenwinkel *et al.* used mixtures of oncogenetic trees [12, 13, 130]. In order to assign positive probabilities to all data points in the input, the topology of the first OT was kept a star tree. For inference of mixtures, they developed an EM-like algorithm, which has not been proven to deliver locally optimal ML solutions.

## 5.2 Hidden-variable Oncogenetic Trees

In Paper IV, we introduce Hidden-variable Oncogenetic Trees (HOTs) and mixtures thereof (HOT-mixtures) in an attempt to remedy some of the problems with traditional OTs, while taking advantage of the simplicity of tree structures.

A HOT is a tree where each vertex is associated with a pair of hidden and visible variables. The hidden variable indicates true progression while the visible variable indicates the outcome of a specific experiment, e.g., the absence or presence of a genetic aberration. The hidden and visible variables associated with a vertex $u$ are denoted $Z(u)$ and $X(u)$, respectively. The values of all variables are assumed to be zero (absence) or one (presence). The distribution on the values of each variable is determined by two conditional probability distributions so that a total of four conditional distributions are associated with each vertex:

$$\begin{aligned}
&\Pr[X(u)|Z(u) = 0],\\
&\Pr[X(u)|Z(u) = 1],\\
&\Pr[Z(u)|Z(p(u)) = 0],\\
&\Pr[Z(u)|Z(p(u)) = 1],
\end{aligned}$$

where $p(u)$ denotes the parent of $u$. Note that the visible variable at a vertex depends only on the hidden variable at the same vertex, and that the hidden variable only depends on the hidden variable of the parent.

When generating a set of aberrations, the values of the hidden variables are determined first. This is similar to oncogenetic trees, except that the hidden variable of a vertex can receive the value one even if its parent has not. The probability $\Pr[Z(u) = 1|Z(p(u)) = 0]$, which is normally small, can be interpreted as the probability that an event associated with a later stage of progression occurs spontaneously although the stages that directly precede it have not been reached. Once the values of the hidden variables are determined, the visible variables receive their

values. The probability $\Pr[X(u) = 1|Z(u) = 0]$ is interpreted as the probability of a false positive and the probability $\Pr[X(u) = 0|Z(u) = 1]$ as a false negative. The latter can also include the probability that the progression has reached $u$ but via a different set of events than the aberration associated with $u$. The set of aberrations generated are the aberrations associated with the vertices whose visible variables have value one.

Paper IV also includes a description of HOT-mixtures. A HOT-mixture consists of a set of HOTs, $\mathcal{T}_1, \ldots, \mathcal{T}_n$, together with a probability distribution on the same. To generate data from a mixture, we first chose a HOT according to the given probability distribution and then generate a set of aberrations from the chosen HOT.

Global structural EM algorithms for inferring HOTs and HOT-mixtures constitute the major computational contributions of Paper IV.

# Chapter 6

# Overview of Included Articles and Manuscripts

**Paper I:** We define a combinatorial model for the reconciliation of gene and species trees using gene duplication, lateral gene transfer, and gene loss. A reconciliation is said to be cyclic if its set of transfers are temporally infeasible. We prove that finding most parsimonious acyclic reconciliations is NP-hard. However, simulations have previously shown that in most cases the most parsimonious reconciliations are acyclic. Dropping the requirement of acyclicity, we provide efficient algorithms for construction of most parsimonious reconciliations. We also analyze a biological dataset with our tools and show that our methods work well in practice.

**Paper II:** We continue to build on the framework provided by our model in Paper I. A thorough discussion on the soundness of our model presented in Paper I is provided. Next, we extend our model to allow arbitrary costs to be associated with duplications and lateral gene transfers and develop efficient methods for finding minimal-cost reconciliations. Analogous to parametric sequence alignment, we derive polynomial-time algorithms for parametric tree reconciliation. Tests are performed on synthetic data that show the performance of our methods.

**Paper III:** Going beyond combinatorial methods, we define a comprehensive probabilistic model of gene evolution that incorporates a birth-death process generating duplications, lateral gene transfers, and losses, together with a substitution model with a relaxed molecular clock. To our knowledge, this is the first probabilistic model used to simultaneously infer duplications and lateral gene transfers, and is more advanced than any probabilistic model that includes LGT as an evolutionary event. We present methods based on MCMC, numerical analysis, and dynamic programming for computing various posterior distributions and probabilities, including the distribution of rates, gene

tree topologies, and counts of lateral gene transfer events.

**Paper IV:** We define Hidden-variable Oncogenetic Trees (HOTs) and mixtures thereof (HOT mixtures) to capture cancer progression pathways. Vertices of a HOT represent specific genetic aberrations and a pair of hidden and visible variables are associated with each vertex. The hidden variables indicate true progression, while the visible variables indicate outcome of experiments for detection of specific aberrations. Global structural EM algorithms are presented for maximum likelihood estimation of HOTs and HOT mixtures from cross sectional data. Analysis of the performance of our methods on synthetic as well as biological data are presented.

# Bibliography


[1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. Garland Science, 2007.

[2] D.J. Araten, D.W. Golde, R.H. Zhang, H.T. Thaler, L. Gargiulo, R. Notaro, and L. Luzzatto. A quantitative measurement of the human somatic mutation rate. *Canc res*, 65(18):8111, 2005.

[3] P. Armitage and R. Doll. The age distribution of cancer and a multi-stage theory of carcinogenesis. *Br J Cancer*, 8(1):1–12, Mar 1954.

[4] L. Arvestad, A.C. Berglund, J. Lagergren, and B. Sennblad. Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. *Proceedings of the eighth annual international conference on research in computational molecular biology*, pages 326–335, 2004.

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

[6] K. Atteson. The performance of neighbor-joining methods of phylogenetic reconstruction. *Algorithmica*, 25(2):251–278, 1999.

[7] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and approximation*. Springer New York, 1999.

[8] O.T. Avery, C.M. MacLeod, and M. McCarty. Chemical nature of the substance inducing transformation of pneumococcal types. induction of transformation by a desoxyribonucleic acid fraction isolated from pneumococcus type III. *J Exp Med*, 79:137–58, 1944.

[9] R.K. Azad and J.G. Lawrence. Detecting laterally transferred genes: use of entropic clustering methods and genome position. *Nucleic Acids Res*, 35(14): 4629–39, 2007.

[10] J.A. Bailey and E.E. Eichler. Primate segmental duplications: crucibles of evolution, diversity and disease. *Nat Rev Genet*, 7(7):552–564, 2006.

[11] E. Bapteste, E. Susko, J. Leigh, D. MacLeod, R.L. Charlebois, and W.F. Doolittle. Do orthologous gene phylogenies really support tree-thinking? *BMC Evol Biol*, 5(1):33, 2005.

[12] N. Beerenwinkel, J. Rahnenfuhrer, M. Daumer, D. Hoffmann, R. Kaiser, J. Selbig, and T. Lengauer. Learning multiple evolutionary pathways from cross-sectional data. *J Comput Biol*, 12(6):584–598, Jul 2005.

[13] N. Beerenwinkel, J. Rahnenfuhrer, R. Kaiser, D. Hoffmann, J. Selbig, and T. Lengauer. Mtreemix: a software package for learning and using mixture models of mutagenetic trees. *Bioinformatics*, 21(9):2106–2107, May 2005.

[14] J.O. Berger and D.A. Berry. Statistical analysis and the illusion of objectivity. *Am Sci*, 76(2):159–165, 1988.

[15] M.J. Bissell and D. Radisky. Putting tumours in context. *Nat Rev Genet*, 1 (1):46–54, 2001.

[16] D. Blackwell and L. Dubins. Merging of opinions with increasing information. *Ann Math Stat*, pages 882–886, 1962.

[17] A. Boc and V. Makarenkov. New efficient algorithm for detection of horizontal gene transfer events. In *Algorithms in Bioinformatics: Third International Workshop, WABI 2003, Budapest, Hungary, September 15-20, 2003: Proceedings*, page 190. Springer, 2003.

[18] C.B. Bridges. Duplication. *Anat Rec*, 15:357–358, 1918.

[19] C.B. Bridges. Salivary chromosome maps. *J Hered*, 26:60–64, 1935.

[20] J.R. Brown. Ancient horizontal gene transfer. *Nat Rev Genet*, 4(2):121–132, 2003.

[21] T.A. Brown. *Genomes*. John Wiley and Sons, Inc., 2002.

[22] F.G. Brunet, H.R. Crollius, M. Paris, J.M. Aury, P. Gibert, O. Jaillon, V. Laudet, and M. Robinson-Rechavi. Gene loss and evolutionary rates following whole-genome duplication in teleost fishes. *Mol Biol Evol*, 23(9):1808–1816, 2006.

[23] A. Caignard, M.S. Martin, M.F. Michel, and F. Martin. Interaction between two cellular subpopulations of a rat colonic carcinoma when inoculated to the syngeneic host. *Int J Canc*, 36(2), 1985.

[24] M.A.A. Castro, T.T.G. Onsten, R.M.C. de Almeida, and J.C.F. Moreira. Profiling cytogenetic diversity with entropy-based karyotypic analysis. *J Theor Biol*, 234(4):487–495, 2005.

[25] L.L. Cavalli-Sforza and A.W. Edwards. Phylogenetic analysis. models and estimation procedures. *Am J Hum Genet*, 19(3 Pt 1):233–257, May 1967.

[26] M.A. Charleston. Jungles: a new solution to the host/parasite phylogeny reconciliation problem. *Math Biosci*, 149(2):191, 1998.

[27] J. Cheetham, F. Dehne, A. Rau-Chaplin, U. Stege, and P.J. Taillon. Solving large FPT problems on coarse-grained parallel machines. *J Comput Syst Sci*, 67(4):691–706, 2003.

[28] T. Clay. Some problems in the evolution of a group of ectoparasites. *Evolution*, pages 279–299, 1949.

[29] P.F. Cliften, R.S. Fulton, R.K. Wilson, and M. Johnston. After the duplication: gene loss and adaptation in Saccharomyces genomes. *Genetics*, 172(2): 863–872, 2006.

[30] B. Crespi and K. Summers. Evolutionary biology of cancer. *Trends Ecol Evol*, 20(10):545–552, 2005.

[31] F.H. Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970.

[32] M. Csűrös and I. Miklós. A probabilistic model for gene content evolution with duplication, loss and horizontal transfer. In *In Tenth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 206–220. Springer, 2006.

[33] B. DasGupta, X. He, T. Jiang, M. Li, and J. Tromp. On the linear-cost subtree-transfer distance between phylogenetic trees. *Algorithmica*, 25(2): 176–195, 1999.

[34] B. DasGupta, X. He, T. Jiang, M. Li, J. Tromp, and L. Zhang. On distances between phylogenetic trees. In *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*, pages 427–436. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 1997.

[35] W.H. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bull Math Biol*, 49(4):461–467, 1987.

[36] P. Dehal and JL Boore. Two rounds of whole genome duplication in the ancestral vertebrate. *PLoS Biol*, 3(10):e314, 2005.

[37] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J Roy Stat Soc Ser B*, pages 1–38, 1977.

[38] R. Desper, F. Jiang, O.P. Kallioniemi, H. Moch, C.H. Papadimitriou, and A.A. Schaffer. Inferring tree models for oncogenesis from comparative genome hybridization data. *J Comput Biol*, 6(1):37–51, Spr 1999.

[39] F.X. Diebold and C. Li. Forecasting the term structure of government bond yields. *J Econometrics*, 130(2):337–364, 2006.

[40] D.S. Dolberg, R. Hollingsworth, M. Hertle, and M.J. Bissell. Wounding and its role in RSV-mediated tumor formation. *Science*, 230(4726):676, 1985.

[41] W.F. Doolittle. Phylogenetic classification and the universal tree. *Science*, 284(5423):2124–9, 1999.

[42] W.F. Doolittle and E. Bapteste. Pattern pluralism and the tree of life hypothesis. *Proc Natl Acad Sci U S A*, 104(7):2043, 2007.

[43] R.G. Downey and M.R. Fellows. *Parameterized complexity*. Springer Verlag, 1999.

[44] R.C. Edgar and S. Batzoglou. Multiple sequence alignment. *Curr Opin Struct Biol*, 16(3):368–373, Jun 2006.

[45] A.W.E. Edwards and L.L. Cavalli-Sforza. The reconstruction of evolution. *Ann Hum Genet*, 27:105–106, 1963.

[46] W. Edwards, H. Lindman, and L.J. Savage. Bayesian statistical inference for psychological research. *Psychol Rev*, 70(3):193–242, 1963.

[47] I. Elias and J. Lagergren. Fast neighbor joining. *Theor Comput Sci*, 410 (21–23):1993–2000, 2009.

[48] J.S. Farris. Inferring phylogenetic trees from chromosome inversion data. *Syst Zool*, 27:275–284, 1978.

[49] M.R. Fellows. On the complexity of vertex set problems. Technical report, Technical report, Computer Science Department, University of New Mexico, 1988.

[50] M.R. Fellows and M.A. Langston. Nonconstructive advances in polynomial-time complexity. *Inf Process Lett*, 26(3):155–162, 1987.

[51] J. Felsenstein. Alternative methods of phylogenetic inference and their interrelationship. *Syst Zool*, 28:49–62, 1979.

[52] J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J Mol Evol*, 17(6):368–376, 1981.

[53] J. Felsenstein. *Inferring phylogenies*. Sinauer Associates Sunderland, 2003.

[54] J. Felsenstein and G.A. Churchill. A hidden Markov model approach to variation among sites in rate of evolution. *Mol Biol Evol*, 13(1):93–104, Jan 1996.

[55] J.A. Fessler and A.O. Hero. Space-alternating generalized expectation-maximization algorithm. *IEEE Trans Signal Process*, 42(10):2664–2677, 1994.

[56] Crick F.H. On protein synthesis. In *Symposia of the Society for Experimental Biology*, volume 12, page 138. Symp Soc Exp Biol, 1958.

[57] W.M. Fitch. Distinguishing homologous from analogous proteins. *Syst Zool*, pages 99–113, 1970.

[58] W.M. Fitch and E. Margoliash. Construction of phylogenetic trees. *Science*, 155(760):279–284, Jan 1967.

[59] A. Force, M. Lynch, F.B. Pickett, A. Amores, Y. Yan, and J. Postlethwait. Preservation of duplicate genes by complementary, degenerative mutations. *Genetics*, 151(4):1531–1545, 1999.

[60] N. Friedman. Learning belief networks in the presence of missing values and hidden variables. In *proc 14th Int Conf on Machine Learning*, page 125. Morgan Kaufmann Pub, 1997.

[61] N. Friedman, M. Ninio, I. Pe'er, and T. Pupko. A structural em algorithm for phylogenetic inference. *J Comp Biol*, 9(2):331–353, 2002.

[62] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman San Francisco, 1979.

[63] O. Gascuel. Concerning the NJ algorithm and its unweighted version, UNJ. In B. Mirkin, F. McMorris, F. Roberts, and A. Rhetsky, editors, *Mathematical Hierarchies and Biology*, pages 149–170. AMS, Providence, 1997.

[64] J.P. Gogarten, W.F. Doolittle, and J.G. Lawrence. Prokaryotic evolution in light of gene transfer. *Mol Biol Evol*, 19(12):2226–2238, 2002.

[65] R. Guigó, I. Muchnik, and T.F. Smith. Reconstruction of ancient molecular phylogeny. *Mol Phylogenet Evol*, 6(2):189–213, 1996.

[66] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.

[67] M. Hallett, J. Lagergren, and A. Tofigh. Simultaneous identification of duplications and lateral transfers. In *Proceedings of the eighth annual international conference on Resaerch in computational molecular biology*, pages 347–356. ACM New York, NY, USA, 2004.

[68] M.T. Hallett and J. Lagergren. New algorithms for the duplication-loss model. *Proceedings of the fourth annual international conference on computational molecular biology*, pages 138–146, 2000.

[69] M.T. Hallett and J. Lagergren. Efficient algorithms for lateral gene transfer problems. *Proceedings of the fifth annual international conference on computational biology*, 2001.

[70] A.T. Hamilton, S. Huntley, M. Tran-Gyamfi, D.M. Baggott, L. Gordon, and L. Stubbs. Evolutionary expansion and divergence in the ZNF91 subfamily of primate-specific zinc finger genes. *Genome Res*, 16(5):584–594, 2006.

[71] D. Hanahan and R.A. Weinberg. The hallmarks of cancer. *Cell*, 100(1):57–70, Feb 2000.

[72] M. Hasegawa, H. Kishino, and T. Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J Mol Evol*, 22(2):160–174, 1985.

[73] W.K. Hastings. Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[74] J. Hein. Reconstructing evolution of sequences subject to recombination using parsimony. *Math Biosci*, 98(2):185–200, 1990.

[75] J. Hein. A heuristic method to reconstruct the history of sequences subject to recombination. *J Mol Evol*, 36(4):396–405, 1993.

[76] J. Hein, T. Jiang, L. Wang, and K. Zhang. On the complexity of comparing evolutionary trees. *Discrete Appl Math*, 71(1-3):153–169, 1996.

[77] A.D. Hershey and M. Chase. Independent functions of viral protein and nucleic acid in growth of bacteriophage. *J Gen Physiol*, 36(1):39–56, 1952.

[78] J.P. Huelsenbeck, F. Ronquist, R. Nielsen, and J.P. Bollback. Bayesian inference of phylogeny and its impact on evolutionary biology. *Science*, 294(5550): 2310–2314, Dec 2001.

[79] M. Jamshidian and R.I. Jennrich. Acceleration of the EM algorithm by using quasi-Newton methods. *J Roy Stat Soc B*, pages 569–587, 1997.

[80] J.A. Jankowski, N.A. Wright, S.J. Meltzer, G. Triadafilopoulos, K. Geboes, A.G. Casson, D. Kerr, and L.S. Young. Molecular evolution of the metaplasia-dysplasia-adenocarcinoma sequence in the esophagus. *Am J Pathol*, 154(4): 965–973, Apr 1999.

[81] E.T. Jaynes. Prior probabilities. *IEEE Trans Syst Sci Cybern*, 227, 1968.

[82] G. Jin, L. Nakhleh, S. Snir, and T. Tuller. Maximum likelihood of phylogenetic networks. *Bioinformatics*, 22(21):2604, 2006.

[83] S. Jones, W.D. Chen, G. Parmigiani, F. Diehl, N. Beerenwinkel, T. Antal, A. Traulsen, M.A. Nowak, C. Siegel, V.E. Velculescu, K.W. Kinzler, B. Vogelstein, J. Willis, and S.D. Markowitz. Comparative lesion sequencing provides insights into tumor evolution. *Proc Natl Acad Sci U S A*, 105(11):4283–4288, Mar 2008.

[84] T.H. Jukes and C.R. Cantor. Evolution of protein molecules. In M.N. Munro, editor, *Mammalian protein metabolism*, volume 3, pages 21–132. New York, 1969.

[85] P.J. Keeling and J.D. Palmer. Horizontal gene transfer in eukaryotic evolution. *Nat Rev Genet*, 9(8):605–618, 2008.

[86] David G. Kendall. On the generalized "birth-and-death" process. *Ann Math Stat*, 19:1–15, 1948.

[87] K.K. Kidd and L.A. Sgaramella-Zonta. Phylogenetic analysis: concepts and methods. *Am J Hum Genet*, 23(3):235–252, May 1971.

[88] M. Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J Mol Evol*, 16(2):111–120, Dec 1980.

[89] H. Kishino and M. Hasegawa. Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in hominoidea. *J Mol Evol*, 29(2):170–179, Aug 1989.

[90] A.G. Knudson, Jr. Mutation and cancer: statistical study of retinoblastoma. *Proc Natl Acad Sci U S A*, 68(4):820–823, Apr 1971.

[91] S. Kumar and A. Filipski. Multiple sequence alignment: in pursuit of homologous DNA positions. *Genome Res*, 17(2):127–135, Feb 2007.

[92] J.G. Lawrence. Horizontal and vertical gene transfer: The life history of pathogens. *Contrib Microbiol*, 12:255–271, 2005.

[93] J. Lederberg and E. Tatum. Gene recombination in Escherichia coli. *Nature*, 158:558, October 1946.

[94] J. Lederberg and E.L. Tatum. Novel genotypes in mixed cultures of biochemical mutants of bacteria. In *Cold Spring Harbor Symp. Quant. Biol*, volume 11, pages 113–114, 1946.

[95] P.M. Lee. *Bayesian Statistics: An Introduction*. John Wiley, 2004.

[96] C. Lengauer, K.W. Kinzler, and B. Vogelstein. Genetic instabilities in human cancers. *Nature*, 396(6712):643–649, 1998.

[97] S. Linz, A. Radtke, and A. von Haeseler. A likelihood framework to measure horizontal gene transfer. *Mol Biol Evol*, 24(6):1312–1319, Jun 2007.

[98] J.S. Liu. *Monte Carlo strategies in scientific computing*. Springer, 2001.

[99] L.A. Loeb. Mutator phenotype may be required for multistage carcinogenesis. *Canc Res*, 51(12):3075–3079, 1991.

[100] M. Lynch and J.S. Conery. The evolutionary fate and consequences of duplicate genes. *Science*, 290(5494):1151–1155, 2000.

[101] M. Lynch and A. Force. The probability of duplicate gene preservation by subfunctionalization. *Genetics*, 154(1):459–473, 2000.

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

[103] C.C. Maley, P.C. Galipeau, J.C. Finley, V.J. Wongsurawat, X. Li, C.A. Sanchez, T.G. Paulson, P.L. Blount, R.A. Risques, P.S. Rabinovitch, *et al.* Genetic clonal diversity predicts progression to esophageal adenocarcinoma. *Nat Genet*, 38(4):468–473, 2006.

[104] C.C. Maley, P.C. Galipeau, X. Li, C.A. Sanchez, T.G. Paulson, P.L. Blount, and B.J. Reid. The combination of genetic instability and clonal expansion predicts progression to esophageal adenocarcinoma. *Canc Res*, 64(20):7629–33, 2004.

[105] C.C. Maley, P.C. Galipeau, X. Li, C.A. Sanchez, T.G. Paulson, and B.J. Reid. Selectively advantageous mutations and hitchhikers in neoplasms: p16 lesions are selected in Barrett's esophagus. *Canc Res*, 64(10):3414, 2004.

[106] T. Marques-Bonet, J.M. Kidd, M. Ventura, T.A. Graves, Z. Cheng, L.D.W. Hillier, Z. Jiang, C. Baker, R. Malfavon-Borja, L.A. Fulton, *et al.* A burst of segmental duplications in the genome of the african great ape ancestor. *Nature*, 457(7231):877–881, 2009.

[107] G. Mendel. Versuche über Pflanzen-Hybriden. *Verb. Naturforsch. Ver. Brunn*, 4:3–47, 1866.

[108] G. Mendel. *Experiments in plant hybridisation*. Cosimo Classics, 2008.

[109] Xiao-Li Meng and David van Dyk. The EM algorithm–an old folk-song sung to a fast new tune. *J Roy Stat Soc Ser B*, 59(3):511–567, 1997.

[110] X.L. Meng and D.B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

[111] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J Chem Phys*, 21 (6):1087–1091, 1953.

[112] A. Meyer and Y. Van de Peer. From 2R to 3R: evidence for a fish-specific genome duplication (FSGD). *Bioessays*, 27(9):937–945, 2005.

[113] F. Michor, Y. Iwasa, and M.A. Nowak. Dynamics of cancer progression. *Nat Rev Genet*, 4(3):197–205, 2004.

[114] B.E. Miller, F.R. Miller, J. Leith, and G.H. Heppner. Growth interaction in vivo between tumor subpopulations derived from a single mouse mammary tumor. *Canc res*, 40(11):3977, 1980.

[115] S.H. Moolgavkar and E.G. Luebeck. Multistage carcinogenesis and the incidence of human cancer. *Gene Chromosome Canc*, 38(4), 2003.

[116] G.W. Moore, M. Goodman, and J. Barnabas. An iterative approach from the standpoint of the additive hypothesis to the dendrogram problem posed by molecular data sets. *J Theor Biol*, 38(3):423, 1973.

[117] H.J. Muller. The origination of chromatin deficiencies as minute deletions subject to insertion elsewhere. *Genetica*, 17(3):237–252, 1935.

[118] H.J. Muller. A viable two-gene deficiency: phenotypically resembling the corresponding hypomorphic mutations. *J Hered*, 26(11):469, 1935.

[119] L. Nakhleh, D. Ruths, and L.S. Want. RIATA-HGT: A fast and accurate heuristic for reconstructing horizontal gene transfer. *Lecture notes in computer science*, pages 84–93, 2005.

[120] R.M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

[121] C.R. Nelson and A.F. Siegel. Parsimonious modeling of yield curves. *J Bus*, pages 473–489, 1987.

[122] C.O. Nordling. A new theory on cancer-inducing mechanism. *Br J Cancer*, 7(1):68–72, Mar 1953.

[123] C. Notredame. Recent evolutions of multiple sequence alignment algorithms. *PLoS Comput Biol*, 3(8):e123, Aug 2007.

[124] P.C. Nowell. The clonal evolution of tumor cell populations. *Science*, 194 (4260):23–28, 1976.

[125] K. Ochiai, T. Yamanaka, K. Kimura, and O. Sawada. Inheritance of drug resistance (and its transfer) between Shigella strains and between Shigella and E. coli strains. *Nihon Iji Shimpo*, 1861:34, 1959.

[126] H. Ochman, J.G. Lawrence, and E.A. Groisman. Lateral gene transfer and the nature of bacterial innovation. *Nature*, 405(6784):299–304, 2000.

[127] S. Ohno. *Evolution by gene duplication*. Allen and Unwin, 1970.

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

[129] R.D.M. Page and M.A. Charleston. Trees within trees: phylogeny and historical associations. *Trends Ecol Evol*, 13(9):356–359, 1998.

[130] J. Rahnenfuhrer, N. Beerenwinkel, W.A. Schulz, C. Hartmann, A. von Deimling, B. Wullich, and T. Lengauer. Estimating cancer survival and clinical outcome based on genetic tumor progression scores. *Bioinformatics*, 21(10): 2438–2446, May 2005.

[131] MJ Renan. How many mutations are required for tumorigenesis? implications from human cancer data. *Mol Carcinog*, 7(3):139, 1993.

[132] F. Ronquist and J.P. Huelsenbeck. Mrbayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*, 19(12):1572–1574, Aug 2003.

[133] A. Rzhetsky and M. Nei. Theoretical foundation of the minimum-evolution method of phylogenetic inference. *Mol Biol Evol*, 10(5):1073–1095, Sep 1993.

[134] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4(4):406–425, Jul 1987.

[135] R.V. Samonte and E.E. Eichler. Segmental duplications and the evolution of the primate genome. *Nat Rev Genet*, 3(1):65–72, 2002.

[136] M. Sanderson and M. McMahon. Inferring angiosperm phylogeny from EST data with widespread gene duplication. *BMC Evol Biol*, 7(Suppl 1):S3, 2007.

[137] D. Sankoff. Minimal mutation trees of sequences. *SIAM J Appl Math*, 28: 35–42, 1975.

[138] D. Sankoff and P. Rousseau. Locating the vertices of a steiner tree in an arbitrary metric space. *Math Program*, 9:240–246, 1975.

[139] B. Sennblad and J. Lagergren. Probabilistic orthology analysis. *submitted*, 2008.

[140] M.H. Sieweke and M.J. Bissell. The tumor-promoting effect of wounding: a possible role for TGF-beta-induced stromal alterations. *Crit Rev Oncog*, 5 (2-3):297, 1994.

[141] P.T. Simpson, J.S. Reis-Filho, T. Gale, and S.R. Lakhani. Molecular evolution of breast cancer. *J Pathol*, 205(2):248–254, Jan 2005.

[142] P.H.A. Sneath and R.R. Sokal. *Numerical taxonomy: The principles and practice of numerical classification*. W.H. Freeman, San Fransisco, 1973.

[143] M.N. Swartz. Use of antimicrobial agents and drug resistance, 1997.

[144] K. Tamura and M. Nei. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Mol Biol Evol*, 10(3):512–526, May 1993.

[145] J.S. Taylor and J. Raes. Duplication and divergence: the evolution of new genes and old ideas. *Annu Rev Genet*, 38:615–43, 2004.

[146] A.A. van Tilborg, A. de Vries, M. de Bont, L.E. Groenfeld, T.H. van der Kwast, and E.C. Zwarthoff. Molecular evolution of multiple recurrent cancers of the bladder. *Hum Mol Genet*, 9(20):2973–2980, Dec 2000.

[147] A. von Haeseler and G.A. Churchill. Network models for sequence evolution. *J Mol Evol*, 37(1):77–85, 1993.

[148] T.L. Wang, C. Rago, N. Silliman, J. Ptak, S. Markowitz, J.K.V. Willson, G. Parmigiani, K.W. Kinzler, B. Vogelstein, and V.E. Velculescu. Prevalence of somatic alterations in the colorectal cancer cell genome. *Proc Natl Acad Sci U S A*, 99(5):3076, 2002.

[149] I. Wapinski, A. Pfeffer, N. Friedman, and A. Regev. Natural history and evolutionary principles of gene duplication in fungi. *Nature*, 449:54–61, 2007.

[150] J.D. Watson and F.H. Crick. Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–8, 1953.

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

[152] D.A. Wheeler, M. Srinivasan, M. Egholm, Y. Shen, L. Chen, A. McGuire, W. He, Y.J. Chen, V. Makhijani, G.T. Roth, *et al.* The complete genome of an individual by massively parallel DNA sequencing. *Nature*, 452(7189): 872–876, 2008.

[153] C.R. Woese. On the evolution of cells. *Proc Natl Acad Sci U S A*, 99(13): 8742–7, 2002.

[154] J. Zhang. Evolution by gene duplication: an update. *Trends Ecol Evol*, 18 (6):292–298, 2003.

[155] Ö. Åkerborg, B. Sennblad, L. Arvestad, and J. Lagergren. Simultaneous Bayesian gene tree reconstruction and reconciliation analysis. *Proc Natl Acad Sci U S A*, 106(14):5714–5719, 2009.

# I

# Simultaneous Identification of Duplications and Lateral Gene Transfers

Ali Tofigh, Michael Hallett, and Jens Lagergren

**Abstract**

The incongruency between a gene tree and a corresponding species tree can be attributed to evolutionary events such as gene duplication and gene loss. This paper describes a combinatorial model where a so-called DTL-scenario is used to explain the differences between a gene tree and a corresponding species tree taking into account gene duplications, gene losses, and lateral gene transfers (also known as horizontal gene transfers). The reasonable biological constraint that a lateral gene transfer may only occur between contemporary species leads to the notion of acyclic DTL-scenarios. Parsimony methods are introduced by defining appropriate optimization problems. We show that finding most parsimonious acyclic DTL-scenarios is NP-complete. However, by dropping the condition of acyclicity, the problem becomes tractable, and we provide a dynamic programming algorithm as well as a fixed-parameter-tractable algorithm for finding most parsimonious DTL-scenarios.

**Index Terms**

Trees, Biology and genetics, Combinatorial algorithms, Graph algorithms.

# Simultaneous Identification of Duplications and Lateral Gene Transfers

## I. Introduction

Gene duplication and lateral gene transfer (LGT) are important evolutionary events that, in interplay with each other as well as with other evolutionary events, shape the genomes of species and thereby also their phenotypes. The role of duplications in creating new functionality has been studied rather extensively. In [1] and [2], the possible fates of a duplicated gene was described by a biological model where the archetypical fates are coined non-functionalization (loss of function due to a disruptive mutation), sub-functionalization (in which the two copies each take on a subset of the original function), and neo-functionalization (where one of the copies, due to point mutations, assumes a new function). Duplications are common in many parts of the tree of life [3], [4], [5], [6], [7]. LGT has also been implicated in how species acquire new functionality and phenotypes. Particular attention has been given to how pathogens have developed through LGT by receiving so called pathogenicity islands, and to the relative importance of LGT and gene loss in pathogen evolution [8]. In contrast to an apparent consensus on the importance of gene duplications, the importance and prevalence of LGT is well known to be unusually controversial (see [9] and references therein).

As always when developing methods for evolutionary studies, there is an interdependence between the underlying models and our knowledge or opinions of the evolutionary processes. When considering LGT, three main views can be identified together with their ramifications for method development. First, one extreme view with few remaining proponents, is that LGT hardly exists, so discrepancies between gene and species trees are due to random effects or insufficiently sophisticated tree reconstruction methods, or possibly due to other events such as duplications. At the other extreme is the view that, at least in some parts of the tree of life, LGT is so rampant that trees are in general not a valid representation of organismal evolution. This latter view is, of course, in conflict with any form of reliance on a species tree when constructing a network or some alternative representation of reticulate evolution caused by LGT. The latter view is, however, fully consistent with the use of gene trees. Finally, there is an intermediate view according to which LGT is common, although not so common among the genes of a species that organismal evolution cannot be meaningfully

described by a tree. When accepting this intermediate view, it becomes desirable to reconstruct species trees, as well as the locations where LGT has occurred, for specific gene families. The parsimony variation of this problem was formalized and treated in [10]. More recently, this approach was applied in an *ad hoc* manner by Bapteste *et al.* [11]. There are also several earlier studies of heuristics for the problem, e.g., [12] and [13], as well as later studies using distance methods [14]. Subsequently, evidence has been provided for the correctness of the intermediate view when considering $\gamma$-proteobacteria [15]. So, a species tree can aid in the identification of LGT and, moreover, species trees requiring fewer LGTs to explain the laterally transferred gene can be viewed as more likely than others.

We may, in general, divide methods attempting to reconstruct evolution into explicit, where a direct interpretation in terms of evolutionary events is possible, and implicit, where no such interpretation is available [16]. We may further distinguish between methods for LGT identification and methods for deriving descriptions of reticulate evolution. Two main methodologies have been applied in order to develop methods for identifying laterally transferred genes. First, atypical sequence characteristics of newly transferred genes have been taken advantage of in order to identify LGTs, see e.g., [17]. Second, incongruencies between gene and species trees have been used in so called phylogenetic LGT identification methods. These methods basically consist of comparing a gene tree with an established species tree and identifying gene tree clades with a significantly different placement in the gene tree compared to the corresponding clade in the species tree. Naive phylogenetic methods are still commonly used.

There has been considerable consensus on the importance and prevalence of duplications, for which the biological model has been significantly clearer. Already in 1979, Goodman *et al.* [18] gave a parsimony method for identification of gene duplications as well as for embedding a gene tree into a corresponding species tree in a way that illustrates a possible evolution of the former *inside* the latter. Building on this framework, Guigo *et al.* [19] gave an explicit supertree method attempting to find the species tree that explains a number of given gene trees using a minimum number of duplications. Ma *et al.* [20] proved hardness results for several variations of species tree reconstruction problems. Hallett *et al.* [21] gave an efficient algorithm that is guaranteed to find the optimum species tree under the assumption that one of the gene trees have had a constant number of lineages in each species tree lineage. More recently, methods have emerged for duplication analysis, i.e., identification of duplications and simultaneous construction of a gene tree, as well as embedding the gene tree inside the species tree. Duplication analysis takes advantage of sequence information directly rather

than merely mediated by a gene tree. In [22], an *ad hoc* method for duplication analysis was presented which also takes gene order information into account. In [23], [24], [25], an integrated model for gene duplication, gene loss, and sequence evolution, together with computational tools for duplication analysis based on the same model, has been developed. In the latest contribution to that line of research, the model was extended with an iid model of sequence evolution rate variation across gene tree edges [26].

There appears to be relatively few studies attempting to tease apart the influences of LGT and gene duplications in regions of the tree of life, where LGT are believed to be common. In [27], it was estimated that 16% of the 1425 intra-genome homologs of *E. coli* K12 have been acquired by LGT, implying that the other homologs have been acquired through gene duplication. An analysis of paralog content in 106 bacterial genomes can be found in [28]. A study by Retchless and Lawrence [29] concluded that the complete separation of *E. coli* and *S. enterica* from their common ancestor took tens of millions of years and that gene conversion events due to bacterial recombination occurred between the incipient species during a period of ~70 million years.

Few attempts have been made at devising methods to explicitly detect duplications and LGTs simultaneously. Csűrös *et al.* [30] gave a probabilistic model of gene evolution that considered LGTs, duplications, and losses, but applied it only to gene family sizes. A rather restricted parsimony method was given in [31] where the input is a gene tree and a species tree augmented with additional edges showing where transfers have taken place. The output is then the minimum cost of mapping the gene tree into the augmented species tree/network.

Here, we present a parsimony method that given a species tree and an incongruent gene tree finds reconciliations that explain the incongruences with a minimum number of duplications and lateral gene transfers. A preliminary version, of which this paper is a complete and thorough revision, first appeared in [32]. In section III, we give the definition of a DTL-scenario (Duplication-Transfer-Loss scenario) which is our formal equivalent of a reconciliation: a description of how a gene tree has evolved within a species tree using, in our case, duplications, LGTs, and losses. Care has been taken in defining DTL-scenarios to include all the interesting viable cases of gene evolution, and at the same time to exclude the cases that seem inappropriate or degenerate in a parsimony setting. Our aim is thus to find DTL-scenarios with a minimum number of duplications and LGTs. As we will demonstrate, the (implicitly inferred) number of losses can be used to choose between several existing most parsimonious DTL-scenarios.

Biologically, LGTs only occur between a pair of contemporary species. It may therefore be desirable to enforce this restriction and demand the existence of a temporal order on the species tree vertices such that all LGTs in the reconciliation occur between contemporary species. We term such DTL-scenarios acyclic. In section V, we show that the problem of finding most parsimonious acyclic DTL-scenarios is NP-complete. However, as was shown in [33], cycles are not a major concern in practice, and in sections VI and VII we provide efficient algorithms for finding most parsimonious DTL-scenarios disregarding the notion of cyclicity. More specifically, in section VI, we provide a dynamic programming algorithm for computing the minimum cost of any DTL-scenario reconciling a gene tree and a species tree, and in section VII, we describe a fixed-parameter-tractable algorithm for enumerating all most parsimonious DTL-scenarios. Finally, in section IX, we demonstrate the benefits of our methods by applying them on real biological data.

But first, we start with a description of the notation that we will use in the remainder of this article.

## II. Definitions

For a directed graph $H$, we let $V(H)$ and $A(H)$ be the sets of vertices and arcs of $H$, respectively. For a tree $T$, we let $V(T)$, $\mathring{V}(T)$, $L(T)$, and $E(T)$ denote the sets of vertices, internal vertices, leaves, and edges of $T$, respectively. For a rooted tree $T$, $\mathrm{root}(T)$ denotes the root vertex. We consider edges of rooted trees to be directed away from the root. An edge of a rooted tree is denoted by an ordered pair of vertices $(u_1, u_2)$ where $u_1$ is closer to the root than $u_2$.

Let $T$ be a rooted tree. If $(u, v)$ is an edge of $T$, then $v$ is called a child of $u$, and $u$ is called the parent of $v$ denoted by $p_T(v)$. When the tree is clear from context, we will drop the subscript and write $p(v)$. Two distinct vertices $u$ and $v$ are siblings iff $p_T(u) = p_T(v)$; in that case, the two edges $(p_T(u), u)$ and $(p_T(v), v)$ are called sibling edges. A vertex $v$ is a descendant of a vertex $u$, denoted by $v \leq_T u$, iff there is a directed path from $u$ to $v$. In that case, we also say that $u$ is an ancestor of $v$ ($u \geq_T v$). We say that $v$ is a proper descendant (proper ancestor) of $u$ iff $v \leq_T u$ ($v \geq_T u$) and $v \neq u$ and denote this by $v <_T u$ ($v >_T u$). An edge whose vertices are ancestors of $v$ is called an ancestral edge of $v$. Two vertices $u$ and $v$ are incomparable iff $u \not\leq_T v$ and $v \not\leq_T u$. An edge $(u, v)$ is called the incoming edge of $v$ and an outgoing edge of $u$. The least common ancestor of a set $X$ of vertices of $T$, denoted $\mathrm{lca}\, X$, is the $\leq_T$-minimal vertex of $T$ that is an ancestor of every vertex in $X$. By

$T_u$ we denote the subtree of $T$ rooted at $u \in V(T)$.

A binary rooted tree is a rooted tree in which every vertex has at most two children. A full binary tree is a rooted tree in which every vertex has zero or two children. A (full rooted binary) forest is a graph in which every connected component is a (full rooted binary) tree. If $T$ is a tree and $F \subseteq E(T)$ is a set of edges of $T$, then $T \setminus F$ is the forest obtained from $T$ by removing the edges in $F$, i.e., $E(T \setminus F) = E(T) \setminus F$.

It will be convenient to describe rooted trees using the Newick format. In this notation, a tree is described using parentheses. If $T_1, \ldots, T_n$ are rooted trees, then $(T_1, \ldots, T_n)$ is the rooted tree obtained from $T_1, \ldots, T_n$ by adding a new root $\rho$ and the edges $(\rho, \mathrm{root}(T_i))$, for $i = 1, \ldots, n$.

Finally, if $f : X \to Y$ is a function from $X$ into $Y$, and if $R \subseteq X$, then the restriction of $f$ to $R$ is denoted by $f|_R$.

## III. DTL-Scenarios

A reconciliation may be thought of as a mapping of a gene tree into a corresponding species tree demonstrating a biologically viable history of the evolution of genes within species. If duplications and losses are the sole culprits in creating incongruencies between the species tree and the gene tree, then there is a unique reconciliation that minimizes both the number of duplications and losses [18] (see also [19], [34], [35]). However, adding LGTs as a possible evolutionary event complicates the notion of a reconciliation; without LGTs, the evolution of the gene tree is conveniently restricted to staying within the edges of the species tree, something that is not true when dealing with LGTs. In fact, when also considering LGTs, there is no simple reconciliation that minimizes the number of evolutionary events. Our approach instead is to define exactly what constitutes a valid reconciliation and devise algorithms that find the most parsimonious ones.

In this section, we define DTL-scenarios (Duplication-Transfer-Loss scenarios) which serve as the formalization of the notion of valid reconciliations. When doing so, we must be careful as to what mappings and combinations of events we wish to allow. We can benefit greatly by carefully defining a reconciliation so as not to allow cases that seem degenerate within a parsimony framework. One example of such a clearly degenerate case is a sequence of LGTs in which the same gene is transferred over and over, and where each transfer is followed by a loss in the species from which the transfer originated. Such a sequence would leave no trace in the intermediate species to which genes have been transferred and would in fact be

represented by a single edge in the gene tree. Such a reconciliation is certainly not interesting in a parsimony setting. Our definition of DTL-scenarios has been carefully crafted to allow biologically viable reconciliations while excluding clearly degenerate cases.

The purpose of a DTL-scenario is to

- assign to each gene tree vertex exactly one event: a speciation, a duplication, or a lateral transfer,
- determine for each transfer vertex exactly one of the outgoing edges as a transfer edge,
- map every vertex of the gene tree into the species tree in a way that is consistent with the previous points and with the temporal order implicitly represented by the trees.

Below, we will formally define a DTL-scenario as an octuple $(S, G, \sigma, \gamma, \Sigma, \Delta, \Theta, \Xi)$. Informally, $S$ and $G$ represent biological data in the form of a species tree and a corresponding gene tree. The correspondence between genes and species is established via a leaf mapping function $\sigma$. Every bifurcation of $G$ is the result of one of three events: speciation, duplication, and lateral transfer; the sets $\Sigma$, $\Delta$, and $\Theta$ contain internal vertices of $G$ representing these events, respectively. The set $\Xi$ contains the edges of $G$ corresponding to lateral transfers. Finally, $\gamma$ maps the entire gene tree into the species tree showing where the evolutionary events have taken place.

Note that $\gamma$ will be defined as a function mapping the gene tree vertices to species tree vertices. For a gene tree vertex $u$, the interpretation of $\gamma(u)$ depends on the type of event represented by $u$ in the DTL-scenario. If $u$ represents a speciation, i.e., $u \in \Sigma$, $\gamma(u)$ is the species tree vertex at which the speciation took place. Otherwise, if $u$ represents a duplication or a lateral transfer, the event represented by $u$ is considered to have occurred somewhere along the incoming edge of $\gamma(u)$ (if $\gamma(u)$ is the root of the species tree, then the event is taken to have occurred before the root). Fig. 1 contains a complete example of a scenario.

Formally, we define a **DTL-scenario** as an octuple $(S, G, \sigma, \gamma, \Sigma, \Delta, \Theta, \Xi)$ where $S$ and $G$ are rooted full binary trees, $\sigma : L(G) \to L(S)$ maps every gene tree leaf to the species in which it is found, $\gamma : V(G) \to V(S)$ maps the gene tree into the species tree, $\Sigma, \Delta$, and $\Theta$ form a partition of $\mathring{V}(G)$, and $\Xi \subset E(G)$ is a subset of the gene tree edges such that:

(I) For each leaf $u$ in the gene tree, $\gamma(u) = \sigma(u)$

(II) If $u \in \mathring{V}(G)$ is a gene tree vertex with children $v$ and $w$, then

a) $\gamma(u)$ is not a proper descendant of $\gamma(v)$ or $\gamma(w)$

b) At least one of $\gamma(v)$ and $\gamma(w)$ is a descendant of $\gamma(u)$

(III) $(u, v) \in \Xi$ if and only if $\gamma(u)$ is incomparable to $\gamma(v)$

(IV) If $u \in \mathring{V}(G)$ is a gene tree vertex with children $v$ and $w$, then

a) $u \in \Theta$ if and only if $(u, v) \in \Xi$ or $(u, w) \in \Xi$

b) $u \in \Sigma$ only if $\gamma(u) = \mathrm{lca}\,\{\gamma(v), \gamma(w)\}$ and $\gamma(v)$ and $\gamma(w)$ are incomparable

c) $u \in \Delta$ only if $\gamma(u) \geq \mathrm{lca}\,\{\gamma(v), \gamma(w)\}$

The cost of a DTL-scenario $\alpha$ is denoted $|\alpha|$, and is defined as $|\Delta| + |\Theta|$ (which is equal to $|\Delta| + |\Xi|$). For convenience, we will allow ourselves to use $\sigma$ as a function mapping a *set* of gene tree leaves to the corresponding *set* of species tree leaves. In this text, the words DTL-scenario and scenario will be used interchangeably.

Condition (I) states that $\gamma$ is an extension of $\sigma$. Condition (IIa) ensures that genes evolve in the direction implied by the trees. Condition (IIb) restricts each bifurcation of $G$ to represent exactly one evolutionary event. Condition (III) determines which edges of the gene tree are to be considered as lateral transfer edges, and condition (IV) states when a gene tree vertex may represent a lateral transfer, speciation, or duplication. Note the overlap between conditions (IVb) and (IVc): given a mapping $\gamma$, the set of gene tree vertices that may be labeled as speciations according to condition (IVb) is a subset of those that may be labeled as duplications according to (IVc). Of course, no most parsimonious DTL-scenario will label a gene tree vertex as a duplication if the vertex may just as well be labeled a speciation. But for now, we will allow this slight over-expressiveness of DTL-scenarios.

For convenience, we will adopt the following notational conventions throughout the paper. The symbols $S$, $G$, $\sigma$, $\gamma$, $\Sigma$, $\Delta$, $\Theta$, $\Xi$, and their subscripted versions, will be used exclusively as the elements of DTL-scenarios. If $\alpha_\Box$ is a DTL-scenario, where $\Box$ is some subscript, then the elements of $\alpha_\Box$ are $S_\Box$, $G_\Box$, $\sigma_\Box$, $\gamma_\Box$, $\Sigma_\Box$, $\Delta_\Box$, $\Theta_\Box$, and $\Xi_\Box$, respectively. If a symbol referring to a scenario lacks subscript, then so will its elements. In that case, it will always be clear from context to which scenario the element symbols belong. The expression "$\alpha_\Box$ is a scenario for $S$, $G$, and $\sigma$" is understood to mean that $S_\Box = S$, $G_\Box = G$, and $\sigma_\Box = \sigma$.

In our scenarios, the interpretation of a transfer edge $(u, v) \in \Xi$ is that a lateral transfer has occurred from the incoming edge of $\gamma(u)$ to some ancestral edge of $\gamma(v)$. For a scenario to be biologically meaningful, we must be able to order the species tree vertices in time in such a way that the incoming edge of $\gamma(u)$ overlaps some ancestral edge of $\gamma(v)$. In fact, if $(u', v')$ is also a transfer edge and $v \geq_G v'$, then we must also ensure that the incoming edge of $\gamma(u)$ overlaps some ancestral edge of $\gamma(v')$. Extending this to include all transfer edges,

we will call a scenario **acyclic** iff

(V) There is a total order $<$ on $V(S)$ such that:

a) if $(x, y) \in E(S)$, then $x < y$

b) if $(u, v), (u', v') \in \Xi$ and $v \geq_G v'$, then $p(\gamma(u)) < \gamma(v')$

See Fig. 1 for an example of a cyclic scenario.

We now present a lemma showing when duplications are forced when considering a mapping of the gene tree into the species tree.

*Lemma 1:* Let $\alpha$ be a scenario for $S$, $G$, and $\sigma$, and let $u \in \mathring{V}(G)$ be a gene tree vertex with children $v$ and $w$.

(a) If $\gamma(v)$ and $\gamma(w)$ are comparable, then $u \in \Delta$.

(b) If $\gamma(u) >_S \text{lca}\,\{\gamma(v), \gamma(w)\}$, then $u \in \Delta$.

*Proof:*

(a) Assume, without loss of generality, that $\gamma(v) \leq_S \gamma(w)$. By (IVb), $u \notin \Sigma$. From (II), we see that $\gamma(u)$ must be an ancestor of both $\gamma(v)$ and $\gamma(w)$. Therefore, by (III) and (IVa), $u \notin \Theta$. By the definition of a scenario, the sets $\Sigma$, $\Delta$, and $\Xi$ partition the internal vertices of $G$. Hence, having shown that $u \notin \Sigma$ and $u \notin \Theta$, we deduce that $u \in \Delta$.

(b) Assume that $\gamma(u) >_S \text{lca}\,\{\gamma(v), \gamma(w)\}$. Since $\gamma(u)$ is comparable to both $\gamma(v)$ and $\gamma(w)$, $u \notin \Theta$. Since $\gamma(u) \neq \text{lca}\,\{\gamma(v), \gamma(w)\}$, $u \notin \Sigma$. Hence, $u \in \Delta$.

■

The minimum number of losses inferred from a scenario can be computed by considering each non-transfer edge $(u, v)$ of the gene tree and the mapping of its vertices into the species tree. This is similar to how losses are computed in the duplication-loss model. One loss is inferred for each intermediate species tree vertex between $\gamma(u)$ and $\gamma(v)$. A loss is also inferred when $u \in \Delta$ and $\gamma(v) \neq \gamma(u)$. We can make this argument formal as follows. Let $\alpha$ be a scenario for $S$, $G$, and $\sigma$ and define $I_\alpha(e)$, where $e = (u, v) \in E(G)$, to be the number of intermediate species tree vertices between $\gamma(u)$ and $\gamma(v)$:

$$I_\alpha(e) = |\{x \in V(S) : \gamma(v) <_S x <_S \gamma(u)\}|.$$

Note that $I_\alpha(e) = 0$ when $e \in \Xi$. The number of losses inferred by $e$ is

$$\text{loss}_\alpha(e) = \begin{cases} I_\alpha(e) + 1 & \text{if } u \in \Delta \text{ and } \gamma(u) \neq \gamma(v) \\ I_\alpha(e) & \text{otherwise.} \end{cases}$$

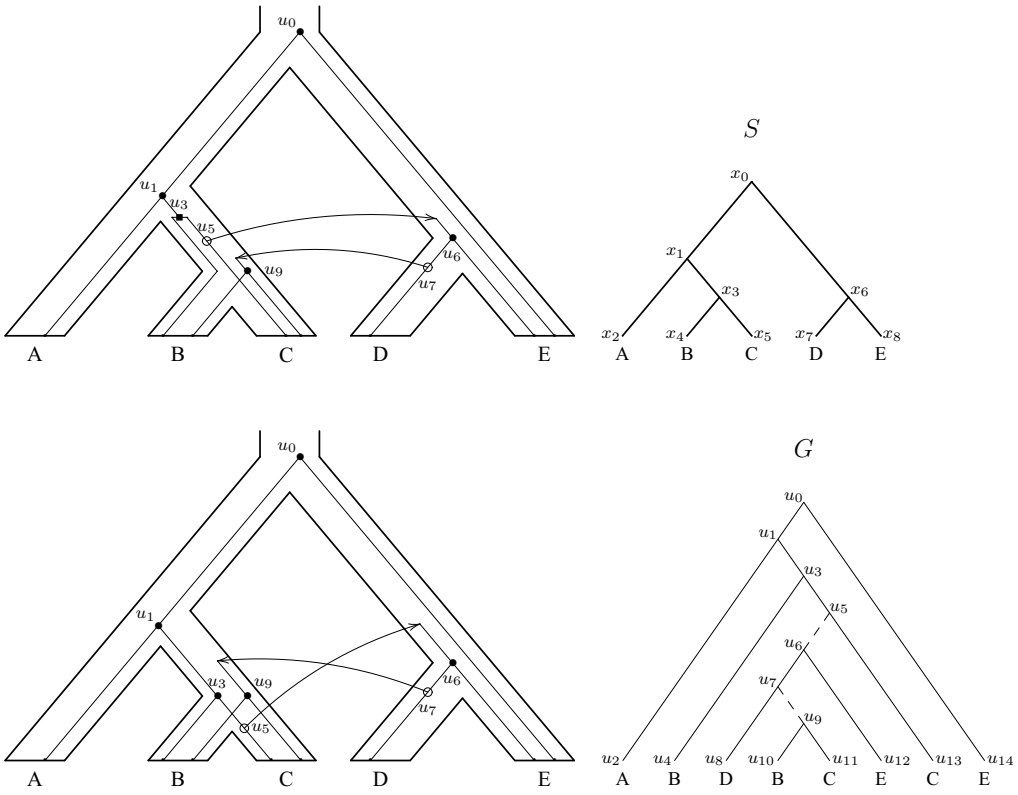



Fig. 1. **An example of a scenario.** A species tree $S$ and a gene tree $G$ are shown on the right side of the figure. The names of the extant species are given below the leaves of $S$. The extant genes of $G$ are labeled with the name of the species to which they belong. The mapping $\sigma$ is then derived from these labels:$\sigma(u_2) = x_2, \sigma(u_4) = x_4, \sigma(u_8) = x_7, \sigma(u_{10}) = x_4, \sigma(u_{11}) = x_5, \sigma(u_{12}) = x_8, \sigma(u_{13}) = x_5$, and $\sigma(u_{14}) = x_8$. A DTL-scenario for $S$, $G$, and $\sigma$ is shown on the top left of the figure. The leaves of $G$ are mapped by $\gamma$ according to $\sigma$. For internal vertices of $G$, we have that $\gamma(u_0) = x_0$, $\gamma(u_1) = x_1$, $\gamma(u_3) = x_3$, $\gamma(u_5) = x_3$, $\gamma(u_6) = x_6$, $\gamma(u_7) = x_7$, and $\gamma(u_9) = x_3$. Two of the edges of the gene tree are transfer edges in this scenario: $\Xi = \{(u_5, u_6), (u_7, u_9)\}$. The sets of speciations, duplications, and transfer vertices are: $\Sigma = \{u_0, u_1, u_6, u_9\}, \Delta = \{u_3\}, \Theta = \{u_5, u_7\}$. It is easy to check that this scenario is acyclic. On the other hand, the scenario depicted in the lower left of the figure is cyclic; we cannot order the species tree vertices in time so that $x_6$ comes before $x_3$ and $x_3$ comes before $x_6$. Note that in this last scenario the least-common-ancestor mapping was used to map $G$ into $S$, which is not the case in the scenario on the top left.

The total number of losses of the scenario is then

$$\text{loss}(\alpha) = \sum_{e \in E(G)} \text{loss}_\alpha(e).$$

## IV. TRANSFER SETS

In this section we will examine the possible mappings of a gene tree into a corresponding species tree. We will characterize the subsets of $E(G)$ that can serve as a set of transfer edges in a DTL-scenario and show how this characterization leads to a complete understanding of all possible mappings. Closely linked to this characterization is a least-common-ancestor mapping that we will define shortly.

As stated earlier, there is a unique mapping of the gene tree into the species tree under the duplication-loss model that simultaneously minimizes the number of duplications and losses. This mapping is defined as

$$M(u) = \mathrm{lca}\,(\sigma(L(G_u))), \tag{1}$$

for all $u \in V(G)$. We now define a similar mapping that will depend, not only on $G$, $S$, and $\sigma$, but also on the set of gene tree edges that we have chosen as transfer edges. Intuitively, given a set $F \subset E(G)$ as the set of transfer edges, we first remove from $G$ all the edges of $F$ to obtain a forest of rooted trees. Each tree in the forest is then mapped into $S$ using (1).

Formally, if $F \subset E(G)$ is a set of gene tree edges such that no two edges of $F$ are siblings, we define the function $\lambda_{G\setminus F} : V(G) \to V(S)$, called the least-common-ancestor mapping of $G$ into $S$, by

$$\lambda_{G\setminus F}(u) = \mathrm{lca}\,(\sigma(L_u)),$$

where $L_u$ is the set of leaves of $G_u$ reachable from $u$ using only edges not in $F$, i.e., only using edges in $G \setminus F$. Note that if $F = \emptyset$, then $\lambda_{G\setminus F} = M$. The next lemma shows that $\lambda_{G\setminus F}$ can be computed recursively in postorder.

*Lemma 2:* Let $S$, $G$, and $\sigma$ be given, and let $F \subset E(G)$ be a set of gene tree edges such that no two edges are siblings. Then, for $u \in V(G)$,

$$\lambda_{G\setminus F}(u) = \begin{cases} \sigma(u) & \text{if } u \in L(G), \\ \mathrm{lca}\,\{\lambda_{G\setminus F}(v), \lambda_{G\setminus F}(w)\} & \text{if } (u,v) \notin F \text{ and } (u,w) \notin F, \text{ where } v, w \text{ are the children of } u, \\ \lambda_{G\setminus F}(w) & \text{if } (u,v) \in F \text{ and } (u,w) \notin F, \text{ where } v, w \text{ are the children of } u, \end{cases}$$

*Proof:* For a vertex $u \in V(G)$, let $L_u$ denote the set of leaves of $G_u$ reachable from $u$ using only edges in $G \setminus F$. The first case follows immediately from the definition of $\lambda_{G\setminus F}$. For the third case, just note that $(u,v) \in F$ implies that $L_u = L_w$. To verify the second case,

note that for vertex sets $A$ and $B$, $\operatorname{lca} A \cup B = \operatorname{lca}\{\operatorname{lca} A, \operatorname{lca} B\}$, so that

$$\begin{aligned}\lambda_{G\setminus F}(u) &= \operatorname{lca}(\sigma(L_u)) \\ &= \operatorname{lca}(\sigma(L_v) \cup \sigma(L_w)) \\ &= \operatorname{lca}\{\operatorname{lca}(\sigma(L_v)), \operatorname{lca}(\sigma(L_w))\} \\ &= \operatorname{lca}\{\lambda_{G\setminus F}(v), \lambda_{G\setminus F}(w)\}.\end{aligned}$$

■

The importance of the least-common-ancestor mapping just defined is highlighted by the next result, which shows that given a DTL-scenario $\alpha$, the lowest possible placement for any gene tree vertex $u$ in the species tree is $\lambda_{G\setminus\Xi}(u)$

*Lemma 3:* If $\alpha$ is a scenario for $S$, $G$, and $\sigma$, then

$$\gamma(u) \geq_S \lambda_{G\setminus\Xi}(u), \tag{2}$$

for any vertex $u \in V(G)$.

*Proof:* Assume $\alpha$ is a scenario for $S$, $G$, and $\sigma$. For $u \in L(G)$, (2) follows immediately from (I) and the definition of $\lambda_{G\setminus\Xi}$.

Let $u \in \mathring{V}(G)$ be a gene tree vertex with children $v$ and $w$ such that $\gamma(u') \geq_S \lambda_{G\setminus\Xi}(u')$ for each proper descendant $u'$ of $u$. If $u \in \Sigma$ or $u \in \Delta$, then

$$\gamma(u) \geq_S \operatorname{lca}\{\gamma(v), \gamma(w)\} \geq_S \operatorname{lca}\{\lambda_{G\setminus\Xi}(v), \lambda_{G\setminus\Xi}(w)\} = \lambda_{G\setminus\Xi}(u),$$

where the first inequality follows from (IVb) and (IVc), the second inequality follows from our inductive hypothesis, and the third equality follows from Lemma 2. Hence, $\gamma(u) \geq_S \lambda_{G\setminus\Xi}(u)$. Assume that $u \in \Theta$ and, without loss of generality, let $(u, v) \in \Xi$. By Lemma 2, $\lambda_{G\setminus\Xi}(u) = \lambda_{G\setminus\Xi}(w)$. Moreover, $\gamma(u)$ must be an ancestor of $\gamma(w)$, and hence,

$$\gamma(u) \geq_S \gamma(w) \geq_S \lambda_{G\setminus\Xi}(w) = \lambda_{G\setminus\Xi}(u).$$

■

Next, we show that it is possible to characterize all subsets $F \subset E(G)$ for which there is a DTL-scenario such that $\Xi = F$, and that for each such $F$, there is a DTL-scenario such that $\Xi = F$ and $\gamma = \lambda_{G\setminus F}$.

A set $F \subset E(G)$ of gene tree edges is called a **transfer set** if no pair of edges in $F$ are siblings and $\lambda_{G\setminus F}(u)$ is incomparable to $\lambda_{G\setminus F}(v)$ for each edge $(u, v) \in F$. We will show that for each transfer set $F$ there is a DTL-scenario such that $\Xi = F$, and that the edges $\Xi$ of

any DTL-scenario is a transfer set. For convenience, we define the concept of anchors, which will also be frequently used later when we discuss our algorithms. We say that $u \in \mathring{V}(G)$ is an **anchor** with respect to a transfer set $F$ iff $\lambda_{G\setminus F}(u) \neq \lambda_{G\setminus F}(v)$ for any child $v$ of $u$. Note that this is equivalent to $\lambda_{G\setminus F}(v)$ being incomparable to $\lambda_{G\setminus F}(w)$ where $v, w$ are the children of $u$. Also note that if $(u, v) \in F$, then $u$ is not an anchor with respect to $F$. See the example on the lower left of Fig. 1 where, in fact, the gene tree is mapped into the species tree using $\lambda_{G\setminus\Xi}$. In the example, the anchors with respect to $\Xi$ are exactly the set of speciations of the gene tree.

*Lemma 4:* Let $S$, $G$ and $\sigma$ be given, and let $F$ be a transfer set. If

$$\Theta = \{u \in \mathring{V}(G) : (u, v) \in F \text{ for some child } v \text{ of } u\},$$

$$\Sigma \subseteq \{u \in \mathring{V}(G) : u \text{ is an anchor w.r.t. } F\}, \quad \text{and}$$

$$\Delta = \mathring{V}(G) \setminus (\Theta \cup \Sigma),$$

then the octuple $(S, G, \sigma, \lambda_{G\setminus F}, \Sigma, \Delta, \Theta, F)$ is a DTL-scenario.

*Proof:* We only need to verify that each requirement of a DTL-scenario is fulfilled. Using Lemma 2, this verification becomes straightforward and is omitted. ■

*Lemma 5:* Let $\alpha$ be a DTL-scenario for $S$, $G$, and $\sigma$. Then $\Xi$ is a transfer set.

*Proof:* assume that $(u, v) \in \Xi$ and $\lambda_{G\setminus\Xi}(u)$ is comparable to $\lambda_{G\setminus\Xi}(v)$. Then, by Lemma 3, $\gamma(u)$ and $\gamma(v)$ must also be comparable, contradicting (III). Hence, $\Xi$ is a transfer set. ■

We now see that the transfer sets induce a natural partition on the space of all DTL-scenarios. Moreover, given a transfer set $\Xi$ we can obtain all possible mappings of $G$ into $S$ by starting with $\lambda_{G\setminus\Xi}$ and placing gene tree vertices closer to the root of $S$ while ensuring that the conditions of a DTL-scenario, especially (II) and (III), are not violated.

In sections VI and VII we will give algorithms for finding scenarios with the least number of duplications and transfers. As we have seen, the transfer sets determine the possible mappings of a gene tree into the species tree, and by using the least-common-ancestor mapping defined above, we can find a mapping that minimizes the number of duplications and losses, just as in the duplication-loss model. Therefore, our intention will not be to find the exact locations within the species tree where events have taken place, but rather to pinpoint what events have taken place in the gene tree.

## V. Finding Most Parsimonious Acyclic Scenarios is NP-Hard

In this section, we will prove that the following decision problem is NP-complete:

DTL-RECONCILIATION

**Instance:** A species tree/gene tree pair $S, G$ with corresponding leaf mapping $\sigma : L(G) \to L(S)$, and a non-negative integer $J \leq |\mathring{V}(G)|$.

**Question:** Is there an acyclic DTL-scenario for $S$, $G$, and $\sigma$ with cost at most $J$?

The NP-completeness will be shown by a reduction from the following NP-complete problem [36]:

MINIMUM FEEDBACK ARC SET

**Instance:** Directed graph $H$ and positive integer $K \leq |A(H)|$.

**Question:** Is there a subset $A' \subseteq A(H)$ with $|A'| \leq K$ such that $A'$ contains at least one arc from every directed cycle in $H$?

Let $H$ and $K$ be given, and let $m = |A(H)|$. We will construct $S$, $G$, and $\sigma$ such that there exists an acyclic DTL-scenario for $S$, $G$, and $\sigma$ with cost at most $J = 2m + K$ if and only if $H$ and $K$ form a yes-instance of MINIMUM FEEDBACK ARC SET.

Let $V = \{r_1, r_2, \ldots, r_n\}$ and $A = \{a_1, a_2, \ldots, a_m\}$ be the sets of vertices and arcs of $H$, respectively. We now give the species tree and gene tree in Newick format. See also Fig. 2 and 3. For each $r_j$, let $S^{r_j}$ be the subtree defined as

$$S^{r_j} = (x_{j,K+6}, (x_{j,K+5}, (\ldots, (x_{j,2}, x_{j,1}) \ldots))).$$

Our species tree is then

$$S = (a_1, (a_2, (\ldots, (a_{m+1}, (S^{r_1}, (S^{r_2}, (\ldots, (S^{r_n}, x_{n+1}) \ldots)))) \ldots))).$$

For each $a_i = \langle r_j, r_k \rangle$, let $G^{a_i}$ be the subtree

$$G^{a_i} = (v^i_{j,K+4}, (v^i_{j,K+3}, (\ldots, (v^i_{j,1}, (v^i_{k,K+6}, v^i_{k,K+5})) \ldots))).$$

Our gene tree is then

$$G = \Big((b_1, G^{a_1}), \Big((b_2, G^{a_2}), \Big(\ldots, \Big((b_m, G^{a_m}), b_{m+1}\Big) \ldots\Big)\Big)\Big).$$

The function $\sigma$ will map the leaves of $G$ to leaves of $S$ according to the subscripts of the leaf labels: $\sigma(v^i_{j,l}) = x_{j,l}$ and $\sigma(b_i) = a_i$.

*Lemma 6:* If $H$ and $K$ form a yes-instance of MINIMUM FEEDBACK ARC SET, then there is a DTL-scenario for $S$, $G$, and $\sigma$ with cost $J = 2m + K$.

*Proof:* Assume that $H$ and $K$ form a yes-instance of MINIMUM FEEDBACK ARC SET, so that there is a subset $A' \subseteq A$, $|A'| = K$, containing at least one arc from every directed

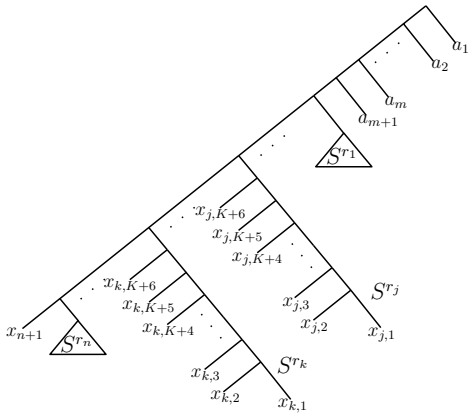



Fig. 2. **The species tree in the NP-completeness proof.** Two subtrees, $S^{r_j}$ and $S^{r_k}$ are shown in full.

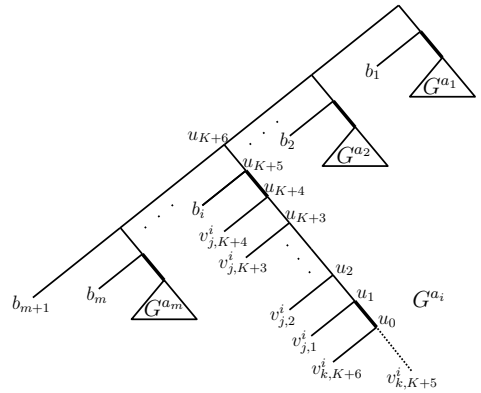



Fig. 3. **The gene tree in the NP-completeness proof.** One subtree $G^{a_i}$ is shown in full, where $a_i = \langle r_j, r_k \rangle$. In Lemma 6, the thick edges are always transfer edges, whereas the dashed edge is a transfer edge iff $a_i \in A'$.

cycle in $H$. We will now prove that there exists an octuple $\alpha = (S, G, \sigma, \gamma, \Sigma, \Delta, \Theta, \Xi)$ that is an acyclic DTL-scenario with cost $J = 2m + K$.

Let $\Xi$ contain the following $J$ edges of $G$:

- For each $a_i = \langle r_j, r_k \rangle$ in $A$, the incoming edge of the root of $G^{a_i}$.
- For each $a_i = \langle r_j, r_k \rangle$ in $A$, the incoming edge of $p(v^i_{k,K+5})$.
- For each $a_i = \langle r_j, r_k \rangle$ in $A'$, the incoming edge of $v^i_{k,K+5}$.

See Fig. 3 where the edges of $\Xi$ are highlighted. Let the sets $\Theta$, $\Sigma$, and $\Delta$ be defined as

$$\Theta = \{u \in \mathring{V}(G) : (u, v) \in \Xi \text{ for some child } v \text{ of } u\},$$

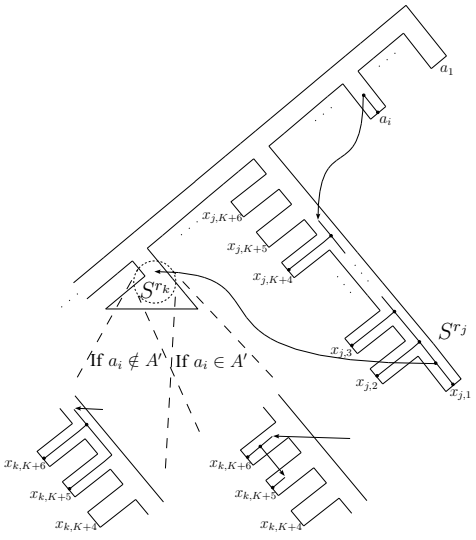



Fig. 4. **Depiction of the mapping of $(b_i, G^{a_i})$ into $S$ where $a_i = \langle r_j, r_k \rangle$.** Note that the mapping differs depending on whether or not $a_i \in A'$.

$$\Sigma = \mathring{V}(G) \setminus \Theta, \quad \text{and}$$
$$\Delta = \emptyset,$$

and let $\gamma = \lambda_{G \setminus \Xi}$. Note that $\Theta$, $\Sigma$, and $\Delta$ partition the elements of $\mathring{V}(G)$, and that we are using the least common ancestor mapping, $\lambda_{G \setminus \Xi}$, to map the gene tree into the species tree. Clearly, no two edges of $\Xi$ are sibling edges, so that $\lambda_{G \setminus \Xi}$ is well defined. See Fig. 4 for an illustration of this mapping.

Intuitively, the part of the species tree containing the leaves $x_{j,K+6}$ and $x_{j,K+5}$ should be thought of as the in-section of $S^{r_j}$, and the part containing $x_{j,K+4}, \ldots, x_{j,1}$ as the out-section of $S^{r_j}$. The reason, as can be seen in Fig. 4, is that we will recreate arcs of $H$ in our scenario by lateral transfers. An arc $\langle r_j, r_k \rangle$ will result in a lateral transfer from the out-section of $S^{r_j}$ to the in-section of $S^{r_k}$.

If we can show that (i) $\Xi$ is a transfer set, and that (ii) for each vertex $u \in \Sigma$, $u$ is an anchor w.r.t. $\Xi$, then we can deduce by Lemma 4 that $\alpha$ is a DTL-scenario.

Consider the subtree $(b_i, G^{a_i})$ for some $a_i = \langle r_j, r_k \rangle \in A$. For ease of notation, let $u_0, u_1, \ldots, u_{K+6}$ denote the internal vertices of the subtree $(b_i, G^{a_i})$ and its parent from bottom up, i.e., let $u_0$ denote $p(v^i_{k,K+5})$ and let $u_l$ denote $p(u_{l-1})$, for $l = 1, \ldots, K+6$. See

Fig. 3.

*Claim 6.1:*

$$\gamma(u_0) = \begin{cases} x_{k,K+6} & \text{if} \quad a_i \in A', \\ p(x_{k,K+6}) & \text{otherwise,} \end{cases}$$

$$\gamma(u_1) = x_{j,1},$$

$$\gamma(u_l) = p(x_{j,l}), \quad \text{for} \quad l = 2, \ldots, K+4,$$

$$\gamma(u_{K+5}) = a_i,$$

$$\gamma(u_{K+6}) = p(a_i).$$

In proving the above claim we will make frequent use of the result presented in Lemma 2 without repeatedly referring to it.

If $a_i \in A'$, then the edge $(u_0, v^i_{k,K+5})$ is in $\Xi$, and by Lemma 2, $\gamma(u_0) = \gamma(v^i_{k,K+6}) = x_{k,K+6}$. If $a_i \notin A'$, then $\gamma(u_0) = \mathrm{lca}\{\gamma(v^i_{k,K+5}), \gamma(v^i_{k,K+6})\} = p(x_{k,K+6})$. By our construction, $(u_1, u_0) \in \Xi$, so that $\gamma(u_1) = \gamma(v^i_{j,1}) = x_{j,1}$. Since $u_l$ is not a transfer vertex for $l = 2, \ldots, K+4$, we can show by induction on $l$ that $\gamma(u_l) = \mathrm{lca}\{\gamma(v^i_{j,l}), \gamma(u_{l-1})\} = p(x_{j,l})$. For $u_{K+5}$, we have that $(u_{K+5}, u_{K+4}) \in \Xi$. Therefore, $\gamma(u_{K+5}) = \gamma(b_i) = a_i$. As a final step consider $u_{K+6}$. Clearly, $u_{K+6} \in \Sigma$ and $\gamma(u_{K+6}) = p(a_i)$. To see this last point, note that if $i = m$, then $\gamma(u_{K+6}) = \mathrm{lca}\{\gamma(b_{m+1}), \gamma(u_{K+5})\} = \mathrm{lca}\{a_{m+1}, a_m\} = p(a_m)$. We can then show by induction on $i$ that $\gamma(u_{K+6}) = p(a_i)$ for $i = m-1, \ldots, 1$. This ends the proof of the above claim, from which the validity of (i) and (ii) follows (see also Fig. 4). Hence, $\alpha$ is DTL-scenario.

Having shown that $\alpha$ is a DTL-scenario, we now move on to showing that $\alpha$ is acyclic. To do this, we will order the vertices of the species tree such that (V) is fulfilled. Let $H'$ be the DAG that is obtained from $H$ by removing the arcs in $A'$, i.e., $V(H') = V$ and $A(H') = A \setminus A'$. Since $H'$ is a DAG, there is a topological sort of the vertices of $H'$. Let us fix a topological sort. We now order the vertices of the species tree as follows. The first vertex is $p(a_1)$, followed by $p(a_2)$, and so on until $p(a_{m+1})$. Next, we have $p(\mathrm{root}(S^{r_1}))$, $p(\mathrm{root}(S^{r_2}))$, and so on until $p(\mathrm{root}(S^{r_n}))$. For the rest of the vertices of $S$, if $r_i$ comes before $r_j$ in the topological sort, then let all internal vertices of $S^{r_i}$ come before the internal vertices of $S^{r_j}$ while respecting the partial order given by the edges of $S$. Last of all come the leaves of $S$ (in any order). We refer to this ordering as $<$.

Condition (Va) is clearly fulfilled. To show that (Vb) holds, we must consider for each

edge $(u, v) \in \Xi$, all edges $(u', v') \in \Xi$ such that $v \geq_G v'$ and show that $p(\gamma(u)) < \gamma(v')$. Note that, in particular, we need to show that $p(\gamma(u)) < \gamma(v)$. By our choice of transfer edges, we only need to consider three edges for each subtree $G^{a_i}$.

So, let $a_i = \langle v_j, v_k \rangle \in A$ and, as before, let $u_0$ denote $p(v^i_{k,K+5})$ and let $u_l$ denote $p(u_{l-1})$, for $l = 1, \ldots, K+6$. The three edges of $G^{a_i}$ that we must consider are $(u_{K+5}, u_{K+4})$, $(u_1, u_0)$, and $(u_0, v^i_{k,K+5})$. By Claim 6.1, $p(\gamma(u_{K+t})) = p(a_i)$ which is an ancestor of $\gamma(u_{K+4})$, $\gamma(u_0)$, and $\gamma(v^i_{k,K+5})$. By our construction of $<$, it follows that (Vb) is satisfied for the edge $(u_{K+5}, u_{K+4})$. Now consider the edge $(u_1, u_0)$. By Claim 6.1, $\gamma(u_1) \in V(S^{r_j})$ and $\gamma(u_0) \in V(S^{r_k})$. In fact, we also have that $p(\gamma(u_1)) \in V(S^{r_j})$. If $a_i \notin A'$, then by our construction, all the internal vertices of $S^{r_j}$ come before the internal vertices of $S^{r_k}$, and therefore, $p(\gamma(u_1)) < \gamma(u_0)$. If, on the other hand, $a_i \in A'$, then $\gamma(u_0) = x_{k,K+6}$ which is a leaf of $S$ and all the leaves come after the internal vertices of $S$ in $<$. Hence, in all cases, $p(\gamma(u_1)) < \gamma(u_0)$. Since $v^i_{k,K+5}$ is mapped by $\gamma$ to a species tree leaf, and all species tree leaves come after the internal vertices in $<$, (Vb) holds for $(u_1, u_0)$, and $(u_0, v^i_{k,K+5})$. Hence, we have shown that $\alpha$ is an acyclic DTL-scenario. ■

This concludes one direction of our NP-completeness proof. It remains for us to show the opposite direction.

*Lemma 7:* If there is an acyclic DTL-scenario for $S$, $G$, and $\sigma$ with cost at most $J$, then $H$ and $K$ form a yes-instance of MINIMUM FEEDBACK ARC SET.

*Proof:* Let $\alpha$ be an acyclic DTL-scenario for $S$, $G$, and $\sigma$, with cost $\leq J$. For most of the remainder of the proof we will consider the subtree $(b_i, G^{a_i})$ for an arbitrary arc $a_i = \langle r_j, r_k \rangle \in A$. As before, let $u_0$ denote the internal vertex $p(v^i_{k,K+5})$ and let $u_l$ denote $p(u_{l-1})$, for $l = 1 \ldots K+6$. As a first step, we show that there is a cost of at least 2 associated with the gene tree vertices $u_1, \ldots, u_{K+6}$.

*Claim 7.1:* At most one of $u_1$ and $u_2$ is in $\Sigma$.

Assume $u_1 \in \Sigma$ and $u_2 \in \Sigma$. Then, by (IVb), $\gamma(u_1)$ must be incomparable to $\gamma(v^i_{j,2}) = x_{j,2}$ and it must be an ancestor of $\gamma(v^i_{j,1}) = x_{j,1}$. There is only one such species tree vertex, namely $x_{j,1}$, which is a leaf. Clearly, by (IVb), $u_1$ cannot be mapped to a species tree leaf if it is a member of $\Sigma$, and we have proved the above claim.

*Claim 7.2:* At most one of $u_{K+5}$ and $u_{K+6}$ is in $\Sigma$.

Assume that $u_{K+5} \in \Sigma$. The children of $u_{K+5}$, $b_i$ and $u_{K+4}$, must be mapped by $\gamma$ to incomparable species tree vertices. Since $b_i$ is a leaf, we know that $\gamma(b_i) = \sigma(b_i) = a_i$. So, $\gamma(u_{K+4})$ must be incomparable to $a_i$. By Lemma 3, $\gamma(u_{K+4}) \geq_S \lambda_{G \setminus \Xi}(u_{K+4})$. But since

$a_l \notin \sigma(L(G_{u_{K+4}}))$ for $l = 1, \ldots, i$, we conclude that $\gamma(u_{K+4}) \neq a_l$ for $l = 1, \ldots, i$. This, together with the fact that $\gamma(u_{K+4})$ is incomparable to $a_i$, implies that

$$\gamma(u_{K+4}) \leq_S p(a_{i+1}),$$

so that

$$\begin{aligned}\gamma(u_{K+5}) &= \mathrm{lca}\,\{\gamma(b_i), \gamma(u_{K+4})\} \\ &= \mathrm{lca}\,\{a_i, \gamma(u_{K+4})\} \\ &= p(a_i).\end{aligned}$$

Now, if $u_{K+6}$ is also in $\Sigma$, then $\gamma(u_{K+6})$ must be incomparable to $\gamma(u_{K+5})$. But the only vertices incomparable to $p(a_i)$ are $a_1, \ldots, a_{i-1}$, which are leaves of $S$. Clearly, $u_{K+6}$ cannot be mapped to a leaf if it is a speciation. Hence, if $u_{K+5} \in \Sigma$, then $u_{K+6} \notin \Sigma$.

Our next claim puts a limit on the number of transfers and duplications among the vertices $u_0, \ldots, u_{K+6}$.

*Claim 7.3:* At most $2+K$ of the vertices $u_0, \ldots, u_{K+6}$ are duplications or transfer vertices.

It follows from the two previous claims that every scenario for $S$, $G$ and $\sigma$ has a minimum cost of $2m$. The result then follows by our assumption that our scenario has cost at most $J = 2m + K$.

Next, we will show that there is a transfer event from $S^{r_j}$ to $S^{r_k}$ corresponding to our arc $a_i = \langle r_j, r_k \rangle \in A$.

Let $u_q$ be the $<_G$-minimal vertex in $\{u_1, \ldots, u_{K+3}\}$ such that $(u_q, v^i_{j,q}) \notin \Xi$. Note that such a vertex exists, otherwise $u_l \in \Theta$ for $l = 1, \ldots, K+3$ contradicting claim 7.3.

*Claim 7.4:* $(u_q, u_{q-1}) \in \Xi$.

Let $x$ be the least common ancestor of the roots $S^{r_j}$ and $S^{r_k}$. We first show that if $u_q \notin \Theta$, then $\gamma(u_q) \geq_S x$, from which it follows that $u_l \in \Delta$ for $l = q+1, \ldots, K+4$ contradicting Claim 7.3.

Assume that $u_q \notin \Theta$. Now, $u_0$ has as children two leaves that are mapped by $\gamma$ to $x_{k,K+5}$ and $x_{k,K+6}$. By (IIb), $\gamma(u_0)$ is an ancestor of at least one of $x_{k,K+5}$ and $x_{k,K+6}$. By the definition of $u_q$, $(u_p, v^i_{j,p}) \in \Xi$ for $p = 1, \ldots, q-1$. By (IIb) and (III), we have that

$$\gamma(u_{q-1}) \geq_S \gamma(u_{q-2}) \geq_S \cdots \geq_S \gamma(u_0).$$

Hence, $\gamma(u_{q-1})$ is an ancestor of at least one of $x_{k,K+5}$ and $x_{k,K+6}$. From (IVb) and (IVc), we see that irrespective of whether $u_q \in \Delta$ or $u_q \in \Sigma$,

$$\gamma(u_q) \geq_S \mathrm{lca}\,\{\gamma(u_{q-1}), \gamma(v^i_{j,q})\} = \mathrm{lca}\,\{\gamma(u_{q-1}), x_{j,q})\}.$$

Since $\gamma(u_{q-1})$ is an ancestor of one of $x_{k,K+5}$ and $x_{k,K+6}$, we see that $\mathrm{lca}\{\gamma(u_{q-1}), x_{j,q})\} \geq_S x$, i.e.,

$$\gamma(u_q) \geq_S x.$$

The sibling of $u_q$, i.e., $v^i_{j,q+1}$, is a leaf and is mapped by $\gamma$ to $x_{j,q+1}$ which is comparable to $x$. We then get from Lemma 1a that $u_{q+1} \in \Delta$. From (IVc) we get that $\gamma(u_{q+1}) \geq x$. We can now show inductively that $u_l \in \Delta$ for $l = q+1, \ldots, K+4$. This together with the fact that $u_l \in \Theta$ for $l = 1, \ldots, q-1$, contradicts Claim 7.3. Therefore, $u_q \in \Theta$, and by the definition of $q$, we must have that $(u_q, u_{q-1}) \in \Xi$. This ends the proof of the above claim.

*Claim 7.5:* $p(\gamma(u_q)) \in \mathring{V}(S^{r_j})$.

Since $u_l \in \Theta$ for $l = 1, \ldots, q$, we deduce from claim 7.3 that $q \leq K+2$ and that at least one vertex among $u_{q+1}, \ldots, u_{K+3}$ is a speciation. Let $u_p$ be the $<_G$-minimal such vertex. We will now show that $\gamma(u_p) = p(x_{j,p})$.

Since $u_p \in \Sigma$, its children, $v_{j,p}$ and $u_{p-1}$, are mapped by $\gamma$ to incomparable species tree vertices. Since $v^i_{j,p}$ is a leaf, we know that $\gamma(v^i_{j,p}) = x_{j,p}$. So, $\gamma(u_{p-1})$ is incomparable to $x_{j,p}$. By Lemma 3, $\gamma(u_{p-1}) \geq_S \lambda_{G\setminus\Xi}(u_{p-1})$. The leaves reachable from $u_{p-1}$ in $G \setminus \Xi$ is a subset of $\{v^i_{j,q}, v^i_{j,q+1}, \ldots, v^i_{j,p-1}\}$, so that

$$\lambda_{G\setminus\Xi}(u_{p-1}) \leq_S \mathrm{lca}\{x_{j,q}, x_{j,q+1}, \ldots, x_{j,p-1}\} = p(x_{j,p-1}).$$

Hence, we have that

$$\gamma(u_p) = \mathrm{lca}\{x_{j,p}, \gamma(u_{p-1})\} = p(x_{j,p}).$$

Now, $\gamma(u_q) \geq_S x_{j,q}$, and by (IIa), $\gamma(u_q)$ is not a proper ancestor of $\gamma(u_p) = p(x_{j,p})$. Hence,

$$x_{j,q} \leq_S \gamma(u_q) \leq_S p(x_{j,p}).$$

Clearly, $p(\gamma(u_q)) \in \mathring{V}(S^{r_j})$, and we have proved the claim.

*Claim 7.6:* if $u_0 \notin \Theta$, then $\gamma(u_{q-1}) \geq_S \mathrm{root}(S^{r_k})$.

Assume that $u_0 \notin \Theta$. By (IVb) and (IVc), we have that

$$\gamma(u_0) \geq \mathrm{lca}\{x_{k,K+5}, x_{k,K+6}\} = \mathrm{root}(S^{r_k}).$$

By the definition of $u_q$, we have that $(u_l, v^i_{j,l}) \in \Xi$ for $l = 1, \ldots, q-1$, so that

$$\gamma(u_{q-1}) \geq_S \gamma(u_{q-2}) \geq_S \cdots \geq_S \gamma(u_0).$$

Hence, $\gamma(u_{q-1}) \geq_S \mathrm{root}(S^{r_k})$.

Since our scenario $\alpha$ is acyclic, there exists a total order on the vertices of $S$ satisfying condition (V). Let $<$ be such an order. By (V), $p(\gamma(u_q)) < \gamma(u_{q-1})$. Since $\gamma(u_{q-1})$ is an ancestor of the root of $S^{r_k}$, we have by (Va), that $p(\gamma(u_{q-1})) < z$ for any vertex $z \in V(S^{r_k})$.

To sum things up, we have shown that for any arc $a_i = \langle r_j, r_k \rangle$ of $A$, if $p(v^i_{k,K+5})$ is not a transfer vertex, then there is a vertex $y \in \mathring{V}(S^{r_j})$ such that $y < z$ for each vertex $z \in V(S^{r_k})$.

All that remains is to construct a subset $A'$ of $A$ such that $|A'| \leq K$, and the graph $H'$, with $V(H') = V$ and $A(H') = A \setminus A'$, is a DAG. We will do this by

$$A' = \{a_i = \langle r_j, r_k \rangle : p(v^i_{k,K+5}) \in \Theta\}.$$

From claims 7.1 and 7.2 it follows that $|A'| \leq K$. Assume that $H'$ contains a directed cycle $c$. For an arc $a_i = \langle r_j, r_k \rangle$ in $c$, we know that there is a vertex $y$ of $\mathring{V}(S^{r_j})$ for which $y < z$ for all $z \in \mathring{V}(S^{r_k})$. But this implies that $<$ is a cyclic order on the vertices of $S$ which is absurd. Hence $H'$ is a DAG. ■

*Theorem 1:* DTL-RECONCILIATION is NP-complete.

*Proof:* This follows immediately from Lemma 6 and 7. ■

The reader may have noticed that the construction of $S$, $G$, and $\sigma$ may leave some of the leaves of $S$ without corresponding leaves in $G$. (i.e., $\sigma(L(G)) \neq L(S)$). This does not contradict any of the conditions of a DTL-scenario. However, one may ask whether the hardness result still holds for the special case when $\sigma(L(G)) = L(S)$. The answer is yes, since we can easily reduce the problem of solving the general case to the special case. In fact, it is easy to verify that if $S'$ is obtained from $S$ by removing all subtrees whose leaves have no corresponding leaves in $G$, then any scenario for $S'$, $G$, and $\sigma$ can easily be extended to a scenario for $S$, $G$, and $\sigma$.

## VI. A DYNAMIC PROGRAMMING ALGORITHM

As we saw in the previous section, finding most parsimonious acyclic scenarios is difficult. However, we also saw that the acyclicity requirement was essential in the NP-completeness proof. We will show in the coming sections, that dropping this requirement makes the problem tractable; we are able to find most parsimonious scenarios in polynomial time if we do not require them to be acyclic. We will return to the problem posed by cycles in section VIII. Unless stated otherwise, we will ignore cycles for the time being.

In this section we will present a dynamic programming algorithm that given $S$, $G$ and $\sigma$, computes the cost of a most parsimonious scenario.

We define a counter $c(u,x)$ as the minimum cost of any scenario for $G_u$ and $S$ such that $u$ is mapped to $x \in V(S)$. This, in turn, is the minimum of the minimum costs of having $u$ mapped to $x$ and $u \in \Sigma$, $u \in \Delta$, and $u \in \Theta$. The counters $c_1$, $c_2$, and $c_3$ given below will represent these three mutually exclusive cases. The recursion is as follows:
If $u \in L(G)$, then

$$c(u,x) = \begin{cases} 0 & \text{if } x = \sigma(u), \\ \infty & \text{otherwise.} \end{cases}$$

If $u \in \mathring{V}(G)$ with children $v$ and $w$, then

$$c(u,x) = \min\{c_1(u,x), c_2(u,x), c_3(u,x)\},$$

where

$$c_1(u,x) = \begin{cases} \min\{c(v,y) + c(w,z) : y \text{ incomparable to } z, \text{ and } \operatorname{lca}\{y,z\} = x\ \} & \text{if } x \in \mathring{V}(S), \\ \infty & \text{otherwise,} \end{cases}$$

$$c_2(u,x) = \min\{1 + c(v,y) + c(w,z) : y \leq_S x,\ z \leq_S x\},$$

$$c_3(u,x) = \min\{1 + c(v,y) + c(w,z) : y \leq_S x \text{ and } z \text{ incomparable to } x\}.$$

The minimum cost of a scenario reconciling $S$ and $G$ is then given by

$$\min_{x \in V(S)} c(\operatorname{root}(G), x).$$

An algorithm for computing the above recursion can easily be implemented to run in time $O(|V(G)| \cdot |V(S)|^2)$. Note that although each minimum in the expressions above is taken over a quadratic number of terms, they can easily be computed in linear time. For example,

$$\begin{aligned} c_2(u,x) &= \min\{1 + c(v,y) + c(w,z) : y \leq_S x, z \leq_S x\} \\ &= \min\{c(v,y) : y \leq_S x\} + \min\{c(w,z) : y \leq_S x\} + 1, \end{aligned}$$

and similarly for the other cases. Algorithm 1 shows explicitly how the recursions may be implemented to achieve the above mentioned time complexity. We will now prove the correctness of our recursion.

For a DTL-scenario $\alpha$ and a gene tree vertex $v \in V(G)$, define the restriction of $\alpha$ to $G_v$ as

$$\alpha|_{G_v} = (S, G_v, \sigma|_{L(G_v)}, \gamma|_{V(G_v)}, \Sigma \cap V(G_v), \Delta \cap V(G_v), \Theta \cap V(G_v), \Xi \cap E(G_v)).$$

**Algorithm 1**: Dynamic programming algorithm.

**Input**: $S$, $G$, and $\sigma$.

**Output**: Minimum cost of any DTL-scenario for $S$, $G$, and $\sigma$.

$c \leftarrow \mathsf{Array}\big[1..|V(G)|, 1..|V(S)|\big]$ initialized to $\infty$
**for** $u \in L(G)$ **do**
    $c(u, \sigma(u)) \leftarrow 0$
**end**
**for** $u \in \mathring{V}(G)$ *in postorder* **do**
    **for** $x \in V(S)$ *in postorder* **do**
        Let $v, w$ be the children of $u$
        **if** $x \in \mathring{V}(S)$ **then**
            Let $y, z$ be the children of $x$
            $c_1 \leftarrow \min\Big(\min_{y' \leq y} c(v, y') + \min_{z' \leq z} c(w, z'), \min_{y' \leq y} c(w, y') + \min_{z' \leq z} c(v, z')\Big)$
        **else**
            $c_1 \leftarrow \infty$
        **end**
        $c_2 \leftarrow \min_{x' \leq x} c(v, x') + \min_{x' \leq x} c(w, x') + 1$
        $c_3 \leftarrow \min\Big(\min_{x' \leq x} c(v, x') + \min_{\substack{x' \not\leq x \\ x' \not\geq x}} c(w, x'), \min_{x' \leq x} c(w, x') + \min_{\substack{x' \not\leq x \\ x' \not\geq x}} c(v, x')\Big)$
        $c(u, x) \leftarrow \min\{c_1, c_2, c_3\}$
    **end**
**end**
**return** $\min_{x \in V(S)} c(\mathrm{root}(G), x)$

It is easy to verify that any restriction of a DTL-scenario is itself a DTL-scenario. The next two lemmas show that we can decompose DTL-scenarios into smaller ones and, given certain natural conditions, we are able to combine smaller DTL-scenarios into larger ones.

*Lemma 8:* Let $S$, $G$, and $\sigma$ be given, and let $v$ and $w$ be the children of $u = \mathrm{root}(G)$. Fix a species tree vertex $x$.

Assume that $\alpha_0$ is a scenario for $S$, $G$, and $\sigma$ such that $\gamma_0(u) = x$, and let $\alpha_1 = \alpha_0|_{G_v}$ and $\alpha_2 = \alpha_0|_{G_w}$.

(a) If $u \in \Sigma_0$, then $\gamma_1(v)$ is incomparable to $\gamma_2(w)$, and $\mathrm{lca}\,\{\gamma_1(v), \gamma_2(w)\} = x$.

(b) If $u \in \Delta_0$, then $\gamma_1(v) \leq_S x$ and $\gamma_2(w) \leq_S x$.

(c) If $u \in \Theta_0$, then $\gamma_1(v) \leq_S x$ and $\gamma_2(w)$ is incomparable to $x$, or $\gamma_2(w) \leq_S x$ and $\gamma_1(v)$ is incomparable to $x$.

*Proof:* If we note that $\gamma_0(v) = \gamma_1(v)$ and $\gamma_0(w) = \gamma_2(w)$, then a, b, and c follow immediately from the definition of DTL-scenarios. ■

*Lemma 9:* Let $S$, $G$, and $\sigma$ be given and let $v$ and $w$ be the children of $u = \mathrm{root}(G)$. Fix a species tree vertex $x$.

Assume that $\alpha_1$ is a DTL-scenario for $S$, $G_v$, and $\sigma|_{L(G_v)}$, and $\alpha_2$ is a DTL-scenario for $S$, $G_w$, and $\sigma|_{L(G_w)}$.

(a) If $\gamma_1(v)$ is incomparable to $\gamma_2(w)$, and $\mathrm{lca}\,\{\gamma_1(v), \gamma_2(w)\} = x$, then there is a DTL-scenario $\alpha_0$ for $S$, $G$, and $\sigma$ such that $\alpha_1 = \alpha_0|_{G_v}$, $\alpha_2 = \alpha_0|_{G_w}$, $\gamma_0(u) = x$, and $u \in \Sigma_0$,

(b) If $\gamma_1(v) \leq_S x$, and $\gamma_2(w) \leq_S x$, then there is a DTL-scenario $\alpha_0$ for $S$, $G$ and $\sigma$ such that $\alpha_1 = \alpha_0|_{G_v}$, $\alpha_2 = \alpha_0|_{G_w}$, $\gamma_0(u) = x$, and $u \in \Delta_0$.

(c) If $\gamma_1(v) \leq_S x$, and $\gamma_2(w)$ is incomparable to $x$, then there is a DTL-scenario $\alpha_0$ for $S$, $G$ and $\sigma$ such that $\alpha_1 = \alpha_0|_{G_v}$, $\alpha_2 = \alpha_0|_{G_w}$, $\gamma_0(u) = x$, and $u \in \Theta_0$.

*Proof:* For (a), let $\alpha_1$ and $\alpha_2$ be given. Assume that $\gamma_1(v)$ is incomparable to $\gamma_2(w)$, and that $\mathrm{lca}\,\{\gamma_1(v), \gamma_2(w)\} = x$. Let $\alpha_0$ be the octuple

$$\alpha_0 = (S, G, \sigma, \gamma_0, \Sigma_1 \cup \Sigma_2 \cup \{u\}, \Delta_1 \cup \Delta_2, \Theta_1 \cup \Theta_2, \Xi_1 \cup \Xi_2),$$

where $\gamma_0 : V(G) \to V(S)$ is an extension of both $\gamma_1$ and $\gamma_2$ with $\gamma_0(u) = x$. It is then straightforward to verify that conditions (I)-(IV) (in particular (IVb)) are fulfilled for $u$. That the conditions are fulfilled for the rest of the gene tree vertices and edges follows from the fact that $\alpha_1$ and $\alpha_2$ are DTL-scenarios.

In the same way (b) and (c) can be proved by instead considering the octuples

$$\alpha_0 = (S, G, \sigma, \gamma_0, \Sigma_1 \cup \Sigma_2, \Delta_1 \cup \Delta_2 \cup \{u\}, \Theta_1 \cup \Theta_2, \Xi_1 \cup \Xi_2), \quad \text{and}$$

$$\alpha_0 = (S, G, \sigma, \gamma_0, \Sigma_1 \cup \Sigma_2, \Delta_1 \cup \Delta_2, \Theta_1 \cup \Theta_2 \cup \{u\}, \Xi_1 \cup \Xi_2 \cup \{(u, w)\}),$$

respectively. ■

Finally, the theorem below proves that the recursions above correctly compute the minimum cost of reconciling a gene tree and species tree.

*Theorem 2:* Let $S$, $G$, and $\sigma$ be given. For $u \in V(G)$ and $x \in V(S)$ define the set

$$A = \{\alpha \text{ a DTL-scenario for } S,\ G_u, \text{ and } \sigma|_{L(G_u)} : \gamma(u) = x\}.$$

Then,

$$c(u,x) = \begin{cases} \infty & \text{if } A = \emptyset, \\ \min\limits_{\alpha \in A} |\alpha| & \text{otherwise.} \end{cases}$$

*Proof:* The theorem is clearly true if $u$ is a leaf.

Assume that $u \in \mathring{V}(G)$ and that the theorem is true for all proper descendants of $u$ for all species tree vertices. Let $v$ and $w$ be the children of $u$ and define the following three sets:

$$A_1 = \{\alpha \in A : u \in \Sigma\},$$

$$A_2 = \{\alpha \in A : u \in \Delta\},$$

$$A_3 = \{\alpha \in A : u \in \Theta\}.$$

*Claim 9.1:*

$$c_1(u,x) = \begin{cases} \infty & \text{if } A_1 = \emptyset, \\ \min\limits_{\alpha \in A_1} |\alpha| & \text{otherwise.} \end{cases}$$

Assume $c_1(u,x) \neq \infty$. Then, by the definition of $c_1(u,x)$, there is a pair $y, z$ of incomparable vertices in $S$ such that $\operatorname{lca}\{y,z\} = x$, $c(v,y) \neq \infty$, and $c(w,z) \neq \infty$. By our inductive hypothesis, this implies that there are scenarios $\alpha_1$ for $S$, $G_v$, $\sigma|_{L(G_v)}$, and $\alpha_2$ for $S$, $G_w$, $\sigma|_{L(G_w)}$, such that $\gamma_1(v) = y$ and $\gamma_2(w) = z$. From Lemma 9a, we then see that $A_1 \neq \emptyset$. Therefore, if $A_1 = \emptyset$, then $c_1(u,x) = \infty$.

Assume $A_1 \neq \emptyset$. Let $\alpha \in A_1$ be a scenario with minimum cost and consider the restrictions $\alpha_1 = \alpha|_{G_v}$ and $\alpha_2 = \alpha|_{G_w}$ of $\alpha$. Clearly, by our inductive hypothesis, $c(v, \gamma_1(v)) \leq |\alpha_1|$. In fact, since $\alpha$ is a minimum-cost scenario in $A$, $c(v, \gamma_1(v)) = |\alpha_1|$. Similarly, $|\alpha_2| = c(w, \gamma_2(w))$. From Lemma 8a, we deduce that

$$c_1(u,x) \leq c(v, \gamma_1(v)) + c(v, \gamma_2(w)) = |\alpha|.$$

Assume that the above inequality is strict, i.e., $c_1(u,x) < |\alpha|$. Then there is a pair of incomparable vertices $y, z$ in $S$ such that $c(v,y) + c(w,z) < |\alpha|$ and $\operatorname{lca}\{y,z\} = x$. By our inductive hypothesis, this implies that there are scenarios $\alpha_3$ for $S$, $G_v$, $\sigma|_{L(G_v)}$, and $\alpha_4$ for $S$, $G_w$, $\sigma|_{L(G_w)}$, such that $\gamma_3(v) = y$, $\gamma_4(w) = z$, and $|\alpha_3| + |\alpha_4| < |\alpha|$. From Lemma 9a, we see that there is a scenario $\alpha_0$ with $\alpha_3 = \alpha_0|_{G_v}$, $\alpha_4 = \alpha_0|_{G_w}$, and $u \in \Sigma_0$. Clearly, $\alpha_0 \in A_1$. The fact that $\alpha_0 \in A_1$ and

$$|\alpha_0| = |\alpha_3| + |\alpha_4| < |\alpha|,$$

produces a contradiction. Therefore, $c_1(u,x) = |\alpha|$, and we have proved the above claim.

The next two claims are stated without proofs as the structure of their proofs are very similar to that of the above claim.

*Claim 9.2:*

$$c_2(u,x) = \begin{cases} \infty & \text{if } A_2 = \emptyset, \\ \min\limits_{\alpha \in A_2} |\alpha| & \text{otherwise.} \end{cases}$$

*Claim 9.3:*

$$c_3(u,x) = \begin{cases} \infty & \text{if } A_3 = \emptyset, \\ \min\limits_{\alpha \in A_3} |\alpha| & \text{otherwise.} \end{cases}$$

The theorem follows immediately from the above claims. ■

### *A. A Dynamic Programming Algorithm for the DT-cost set problem*

In the previous section we saw how to compute the minimum cost of reconciling $S$, $G$, and $\sigma$. It can, however, be desirable to know the number of duplications and transfers that are involved in an optimal scenario separately. We can use the same recursive idea as in the previous section to find all pairs $(d,t)$ such that $d$ is the number of duplications and $t$ is the number of transfers in some optimal scenario. Instead of using a counter to keep track of the minimal cost, we will use sets containing pairs of numbers. The recursion is given below without proof. Note that we define $\operatorname{argmin}_{x \in X} f(x)$ to be the *set* of arguments where $f(x)$ attains its minimum value. Also, similar to the counters $c_1$, $c_2$, and $c_3$ above, the sets $A_1$, $A_2$, and $A_3$ defined below correspond to costs of the gene tree vertex under consideration being a speciation, duplication, and transfer vertex respectively. An algorithm for computing the recursion below can easily be implemented to run in time $O(mn(m+n^2))$, where $m = |V(S)|$ and $n = |V(G)|$.

If $u \in L(G)$

$$c(u,x) = \begin{cases} \{(0,0)\} & \text{if } x = \sigma(u) \\ \emptyset & \text{otherwise.} \end{cases}$$

If $u \in \mathring{V}(G)$

$$c(u,x) = \operatorname*{argmin}_{(d,t) \in A_1 \cup A_2 \cup A_3} d + t,$$

where the sets $A_1$, $A_2$, and $A_3$ are defined as:

$$A_1 = \{(d_1 + d_2, t_1 + t_2) : (d_1, t_1) \in c(v, y), (d_2, t_2) \in c(w, z),$$
$$\text{where } y \text{ is incomparable to } z, \text{ and } \mathrm{lca}\,\{y, z\} = x\},$$
$$A_2 = \{(d_1 + d_2 + 1, t_1 + t_2) : (d_1, t_1) \in c(v, y), (d_2, t_2) \in c(w, z),$$
$$\text{where } y \leq_S x, \text{ and } z \leq_S x\},$$
$$A_3 = \{(d_1 + d_2, t_1 + t_2 + 1) : (d_1, t_1) \in c(v, y), (d_2, t_2) \in c(w, z),$$
$$\text{where } y \leq_S x \text{ and } z \text{ incomparable to } x\}.$$

## VII. A Fixed-Parameter-Tractable Algorithm

In this section we present a fixed-parameter-tractable algorithm for reconciling $S$, $G$, and $\sigma$, which is able to enumerate all optimal reconciliations. The time complexity of the algorithm will be shown to be polynomial in the size of the input when considering the minimum cost of reconciling $S$ and $G$ as a fixed parameter.

As we saw in section IV, the transfer sets induce a natural partition of the space of DTL-scenarios, and the main idea behind the FPT-algorithm is to use transfer sets as a basis for searching in this space. We know that there is a DTL-scenario for each transfer set, but every transfer set also entails certain restrictions as to the placement of gene tree vertices within the species tree as seen in Lemma 3. This, in turn, forces the introduction of duplications by Lemma 1. Hence, choosing to include or not to include an edge in a transfer set has consequences in terms of forced duplications. Below, we will define the notion of candidates which will be central to our search strategy. The purpose of a candidate is to keep track of forced duplications with respect to transfer sets and anchors. We remind the reader that a gene tree vertex is called an anchor w.r.t. a transfer set $F$ if its children are mapped to incomparable species tree vertices by $\lambda_{G\setminus F}$. It can then be seen from (IVb) and Lemma 4 that for each transfer set $F$, there is a DTL-scenario in which all anchors w.r.t. $F$ are speciations.

A tuple $(D, F)$ is called a **candidate** for $S$, $G$, and $\sigma$ iff $D \subseteq \mathring{V}(G)$, $F$ is a transfer set, and for each $u \in D$, $(u, v) \notin F$ for any child $v$ of $u$. An internal gene tree vertex $u$ is **unmarked** with respect to a candidate $(D, F)$ iff $u \notin D$ and $(u, v) \notin F$ for any child $v$ of $u$. A candidate $(D, F)$ is **final** iff each unmarked vertex $u$ is an anchor w.r.t. $F$. We will say that $u$ is an anchor w.r.t. a candidate $(D, F)$, when and only when $u$ is an anchor w.r.t. $F$. The cost of a candidate $(D, F)$ is defined as $|(D, F)| = |D| + |F|$. A final candidate with

minimal cost is called **optimal**. Finally, we write $(D_1, F_1) \preceq (D_2, F_2)$ when $D_1 \subseteq D_2$ and $F_1 \subseteq F_2$.

We now show that $(D, F)$ is a final candidate if and only if there is a DTL-scenario such that $D = \Delta$ and $F = \Xi$. It then follows that given any final candidate $(D, F)$, there is a set of corresponding scenarios $\{\alpha : \Delta = D, \Xi = F\}$ in which the only difference between two distinct scenarios is the mapping of the gene tree into the species tree. There is, in general, an exponential number of ways to map a gene tree into a species tree even when keeping $\Delta$, $\Xi$, and $\Sigma$ fixed.

*Lemma 10:* If $(D, F)$ is a final candidate for $S$, $G$, and $\sigma$, then there is a DTL-scenario $\alpha$ for $S$, $G$, and $\sigma$ such that $\Delta = D$ and $\Xi = F$.

*Proof:* Assume that $(D, F)$ is a final candidate for $S$, $G$, and $\sigma$. Let

$$\Theta = \{u \in \mathring{V}(G) : (u, v) \in F \text{ for some child } v \text{ of } u\},$$

$$\Delta = D, \qquad \text{and}$$

$$\Sigma = \{u \in \mathring{V}(G) : u \notin \Theta, u \notin \Delta\}.$$

For each $u \in \Sigma$, $u$ is unmarked w.r.t. $(D, F)$, and since $(D, F)$ is final, $u$ is an anchor w.r.t. $F$. By Lemma 4, we see that the octuple $(S, G, \sigma, \lambda_{G \setminus F}, \Sigma, \Delta, \Theta, F)$ is a DTL-scenario. ■

*Lemma 11:* If $\alpha$ is a scenario for $S$, $G$, and $\sigma$, then $(\Delta, \Xi)$ is a final candidate for $S$, $G$, and $\sigma$.

*Proof:* Assume that $\alpha$ is a scenario for $S$, $G$, and $\sigma$. Since $\Delta \cap \Theta = \emptyset$ and $\Xi$ is a transfer set, we have that $(\Delta, \Xi)$ is a candidate. Clearly, $\Sigma$ is the set of unmarked vertices of $(\Delta, \Xi)$. Let $u \in \Sigma$ with children $v$ and $w$. We need to show that $u$ is an anchor w.r.t. $\Xi$. From Lemma 3, we have that $\lambda_{G \setminus \Xi}(v) \leq_S \gamma(v)$ and $\lambda_{G \setminus \Xi}(w) \leq_S \gamma(w)$. Therefore, since $\gamma(v)$ is incomparable to $\gamma(w)$ by (IVb), $\lambda_{G \setminus \Xi}(v)$ is incomparable to $\lambda_{G \setminus \Xi}(w)$ and $u$ is an anchor w.r.t. $\Xi$. Hence, $(\Delta, \Xi)$ is a final candidate. ■

As stated earlier, our main interest is not determining the exact location within the species tree where events have taken place, but rather to identify the set of duplications and transfers in the gene tree by finding parsimonious scenarios. Therefore, we define an equivalence relation on the set of scenarios such that two scenarios $\alpha_1$ and $\alpha_2$ for $S$, $G$, and $\sigma$ are equivalent iff $\Sigma_1 = \Sigma_2$, $\Delta_1 = \Delta_2$, and $\Xi_1 = \Xi_2$. It is then clear that every final candidate can be taken as the representative of an equivalence class of scenarios.

In the remainder of this section we will give an algorithm that, given $S$, $G$, and $\sigma$, enumerates all optimal candidates. We can think of this computation as search for optimal

candidates in the space of candidates for $S$, $G$, and $\sigma$. Consider a search tree in the space of all candidates where the root is the empty candidate, $(\emptyset, \emptyset)$, and each candidate $C$ has as its children exactly the candidates $C'$ such that $C \preceq C'$ and $|C'| = |C| + 1$ (so each child of $C$ is obtained by adding exactly one duplication or one transfer to $C$). We will show that we can prune this search tree in such a way that no candidate has more than three children and that all optimal candidates are reachable from the root. The algorithm we present below performs an implicit breadth-first search in this "pruned" search tree.

Let $C = (D, F)$ be a candidate for $S$, $G$, and $\sigma$ and let $f$ be the set of transfer vertices of $C$, i.e., $f = \{u \in V(G) : (u, v) \in F \text{ for some child } v \text{ of } u\}$. Consider the forest $G \setminus F$. The vertices with only one outgoing edge in $G \setminus F$ are exactly the set of transfer vertices of $C$. We will often need to speak of pairs of gene tree vertices $u \notin f, v \notin f$, such that $u$ is a proper ancestor of $v$ in $G \setminus F$ and for each vertex $w$ such that $u >_G w >_G v$, $w$ is a transfer vertex. For ease of notation, we will now introduce the rooted forest $G \natural F$ that is obtained from $G \setminus F$ by contracting any paths that contain only transfer vertices into a single edge. Note that $(u, v)$ is an edge of $G \natural F$ iff $u$ is a proper ancestor of $v$ in $G \setminus F$ and every vertex on the path from $u$ to $v$ that is distinct from $u$ and $v$ is a transfer vertex. This implies that if $(u, v)$ is an edge of $G \natural F$ and $u >_G w >_G v$, then $\lambda_{G\setminus F}(w) = \lambda_{G\setminus F}(v)$. Also, note that $G \natural F$ is a full rooted binary forest, i.e., every vertex of $G \natural F$ has two outgoing edges in $G \natural F$.

Let $C = (D, F)$ be a candidate for $S$, $G$, and $\sigma$. An unmarked anchor $u$ w.r.t. $C$ is called an **s-move** iff $(p, u) \in E(G \natural F)$ for some unmarked vertex $p$ and $\lambda_{G\setminus F}(p) = \lambda_{G\setminus F}(u)$. An unmarked vertex $u$ of $C$ is called a **d-move** iff $(u, v) \in E(G \natural F)$ for some vertex $v \in D$ and $\lambda_{G\setminus F}(u) = \lambda_{G\setminus F}(v)$. As we will see, we must decide for each s-move of a candidate $C$ whether it is to become a speciation, in which case its parent in $G \natural F$ is a forced duplication, or to have it be a transfer vertex. A d-move is simply a vertex that must be a duplication due to the choices made so far. Given these definitions, we will now show that Algorithm 2 enumerates all optimal candidates and can be implemented to run in time $O(m + n \cdot 3^c)$, where $n = |V(G)|$, $m = |V(S)|$, and $c$ is the minimum cost of any final candidate for $S$, $G$, and $\sigma$. Therefore, keeping the cost $c$ fixed, the algorithm runs in polynomial time in $m$ and $n$.

First, we give two technical lemmas that will be needed later. Then we will show that a candidate is final iff it contains no d-moves or s-moves. This is the main idea behind Algorithm 2; we identify d-moves and s-moves and eliminate them by introducing duplica-

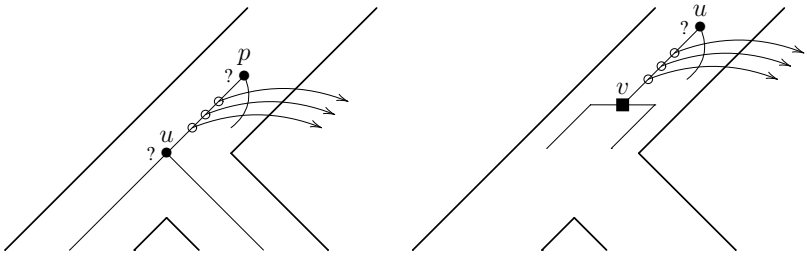



Fig. 5. **Illustration of the d- and s-moves of a candidate** $C = (D, F)$**.** The left figure shows a portion of the gene tree mapped inside the species tree using $\lambda_{G\setminus F}$. The vertex $u$ is an anchor, and both $u$ and $p$ are unmarked and mapped to the same species tree vertex. The vertices in the path between $p$ and $u$ are all transfer vertices of $C$. As proved in the text, every optimal final candidate $C' \succeq C$ either has one of the outgoing edges of $u$ as a transfer edge, or $u$ is unmarked and $p \in D'$. The right figure illustrates a d-move. Both $u$ and $v$ are mapped to the same vertex by $\lambda_{G\setminus F}$, $v \in D$, and $u$ is unmarked. Every optimal final candidate $C' \succeq C$ will have $u \in D'$.

tions and transfers. Note that when eliminating a move, at most three candidates need to be considered. As we will show further below, the order in which we consider the moves is irrelevant.

*Lemma 12:* A gene tree vertex $u$ is an anchor w.r.t. a candidate $(D, F)$ iff $\lambda_{G\setminus F}(u) \neq \lambda_{G\setminus F}(v)$ for each edge $(u, v)$ of $G \setminus\!\!\setminus F$.

*Proof:* Let $C = (D, F)$ be a candidate. Assume $u$ is an anchor w.r.t. $C$ and let $(u, v)$ be an edge of $G \setminus\!\!\setminus F$. Consider the child $v'$ of $u$ that is an ancestor of $v$. By the definition of $G \setminus\!\!\setminus F$, we have that $\lambda_{G\setminus F}(v') = \lambda_{G\setminus F}(v)$. Since $u$ is an anchor, $\lambda_{G\setminus F}(v') \neq \lambda_{G\setminus F}(u)$, and hence, $\lambda_{G\setminus F}(v) \neq \lambda_{G\setminus F}(u)$.

Assume that $\lambda_{G\setminus F}(u) \neq \lambda_{G\setminus F}(v)$ for an edge $(u, v)$ of $G \setminus\!\!\setminus F$. Consider the child $v'$ of $u$ that is an ancestor of $v$. By the definition of $G \setminus\!\!\setminus F$, $\lambda_{G\setminus F}(v') = \lambda_{G\setminus F}(v) \neq \lambda_{G\setminus F}(u)$. Hence, if $u$ is a vertex of $G \setminus\!\!\setminus F$ such that $\lambda_{G\setminus F}(u) \neq \lambda_{G\setminus F}(v)$ for each edge $(u, v)$ of $G \setminus\!\!\setminus F$, then $\lambda_{G\setminus F}(v') \neq \lambda_{G\setminus F}(u)$ for each child $v'$ of $u$, and $u$ is an anchor. ■

*Lemma 13:* Let $C = (D, F)$ and $C' = (D', F')$ be candidates for $S$, $G$, and $\sigma$ such that $C \preceq C'$. We then have that

(a) $\lambda_{G\setminus F'}(u) \leq_S \lambda_{G\setminus F}(u)$ for any gene tree vertex $u$.

(b) If $u$ is unmarked in $C$, but is a transfer vertex in $C'$, then $\lambda_{G\setminus F'}(u) <_S \lambda_{G\setminus F}(u)$.

(c) If $\lambda_{G\setminus F}(p_G(u)) = \lambda_{G\setminus F}(u)$ and $\lambda_{G\setminus F'}(u) = \lambda_{G\setminus F}(u)$, then $\lambda_{G\setminus F'}(p_G(u)) = \lambda_{G\setminus F}(p_G(u))$.

(d) If $\lambda_{G\setminus F'}(u) \neq \lambda_{G\setminus F}(u)$, then there is an anchor $v$ w.r.t. $C$ such that $\lambda_{G\setminus F}(v) = \lambda_{G\setminus F}(u)$

**Algorithm 2**: Fixed-parameter-tractable algorithm.

**Input**: $S$, $G$, and $\sigma$.

**Output**: All optimal final candidates for $S$, $G$, and $\sigma$.

```
1  opt ← ∞
2  Q ← {(∅, ∅)} ;                                   // Initialize queue
3  while Q ≠ ∅ do
4      (D, F) ← dequeue(Q)
5      if |(D, F)| ≤ opt then
6          if there is an s-move u in (D, F) then
7              let v, w be the children of u
8              C1 ← (D, F ∪ {(u, v)})
9              C2 ← (D, F ∪ {(u, w)})
10             C3 ← (D ∪ {p_{G\F}(u)}, F)
11             Q ← Q ∪ {C1, C2, C3}
12         else if there is a d-move u in (D, F) then
13             C1 ← (D ∪ {u}, F)
14             Q ← Q ∪ {C1}
15         else
16             solutions ← solutions ∪ {(D, F)}
17             opt ← |(D, F)|
18         end
19     end
20 end
21 return solutions
```

and $v$ is a transfer vertex in $C'$.

*Proof:*

(a) Since $C \preceq C'$, we have that $F \subseteq F'$ and the result follows from Lemma 2.

(b) Assume that $u$ is unmarked in $C$ but is a transfer vertex in $C'$. Let $(u, v) \in F$ for some child $v$ of $u$. From (a), we already know that $\lambda_{G\setminus F'}(u) \leq_S \lambda_{G\setminus F}(u)$. Since $u$ is unmarked in $C$, $\lambda_{G\setminus F}(u)$ is comparable to $\lambda_{G\setminus F}(v)$. From (a), we have that $\lambda_{G\setminus F}(v) \leq_S \lambda_{G\setminus F'}(v)$. Therefore, if $\lambda_{G\setminus F'}(u) = \lambda_{G\setminus F}(u)$, then $\lambda_{G\setminus F'}(u)$ is comparable to $\lambda_{G\setminus F'}(v)$ so that $F'$

is not a transfer set. Hence, $\lambda_{G\backslash F'}(u) \neq \lambda_{G\backslash F}(u)$.

(c) Assume that $\lambda_{G\backslash F}(p_G(u)) = \lambda_{G\backslash F}(u)$ and $\lambda_{G\backslash F'}(u) = \lambda_{G\backslash F}(u)$. We see from (a) that $\lambda_{G\backslash F'}(p_G(u)) \leq_S \lambda_{G\backslash F}(p_G(u)) = \lambda_{G\backslash F}(u)$. Since $F'$ is a transfer set, it follows from Lemma 2 that $\lambda_{G\backslash F'}(p_G(u))$ is not a proper descendant of $\lambda_{G\backslash F'}(u) = \lambda_{G\backslash F}(u)$. Hence $\lambda_{G\backslash F'}(p_G(u)) = \lambda_{G\backslash F}(u) = \lambda_{G\backslash F}(p_G(u))$.

(d) Assume that $\lambda_{G\backslash F'}(u) \neq \lambda_{G\backslash F}(u)$. Let $u = u_1, u_2, \ldots, u_n$ be a path in $G$ such that $\lambda_{G\backslash F}(u_i) = \lambda_{G\backslash F}(u)$ and $u_n$ is an anchor w.r.t. $C$. If $u_n$ is not a transfer vertex in $C'$, then $\lambda_{G\backslash F'}(u_n) = \lambda_{G\backslash F}(u_n)$. From (c), we see that $\lambda_{G\backslash F'}(u_i) = \lambda_{G\backslash F}(u_i)$ for $i = 1, \ldots, n$ which is a contradiction. Hence, $u_n$ is a transfer vertex in $C'$.

■

Using the rather technical lemmas above, we can now show that final candidates correspond exactly to candidates with no moves.

*Lemma 14:* $C$ is a final candidate iff $C$ is a candidate with no d-moves or s-moves.

*Proof:* Assume that $C$ is final. Each unmarked vertex of $C$ is an anchor w.r.t. $C$. If $u$ is a d-move in $C$, then there is a vertex $v$ such that $(u, v) \in E(G \natural F)$ and $\lambda_{G\backslash F}(v) = \lambda_{G\backslash F}(u)$, and by Lemma 12, $u$ is not an anchor. Hence, there are no d-moves in $C$. If $u$ is an s-move in $C$, there is an edge $(p, u) \in E(G \natural F)$ such that $p$ is unmarked and $\lambda_{G\backslash F}(p) = \lambda_{G\backslash F}(u)$, and by Lemma 12, $p$ is not an anchor. Hence, there are no s-moves in $C$.

For the other direction, assume that $C = (D, F)$ is a candidate with no d-moves or s-moves. Assume that there is an unmarked vertex in $C$ that is not an anchor and let $u$ be a $<_G$-minimal such vertex. By Lemma 12, there is an edge $(u, v)$ in $G \natural F$ such that $\lambda_{G\backslash F}(u) = \lambda_{G\backslash F}(v)$. Clearly, $v$ is not a transfer vertex of $C$. If $v$ is unmarked in $C$, then by the $<_G$-minimality of $u$, $v$ is an anchor, implying that $v$ is an s-move, producing a contradiction. If $v \in D$, then $u$ is a d-move which is also a contradiction. Since we get a contradiction in all cases, we must have that each unmarked vertex of $C$ is an anchor, i.e., $C$ is final. ■

The next two lemmas show that there are three mutually exclusive ways to resolve an s-move. Either the s-move is kept as a speciation, in which case its parent in $G \natural F$ must be labeled a duplication, or one of its outgoing edges is added as a transfer edge.

*Lemma 15:* Let $C = (D, F)$ be a candidate for $S$, $G$, and $\sigma$. If $u$ is an s-move in $C$ with children $v$ and $w$, then for any optimal final candidate $C^* = (D^*, F^*)$ such that $C \preceq C^*$

(a) $u \notin D^*$, and

(b) if $u$ is not a transfer vertex in $C^*$, then $p_{G\natural F}(u) \in D^*$.

*Proof:* Assume that $u$ is an s-move in $C$ with children $v, w$ and let $C^* = (D^*, F^*)$ be an optimal candidate such that $C \preceq C^*$. Let $p = p_{G\setminus F}(u)$.

By the definition of an s-move, $u$ is an anchor w.r.t. $C$ implying that $\lambda_{G\setminus F}(v)$ is incomparable to $\lambda_{G\setminus F}(w)$. By Lemma 13a, $\lambda_{G\setminus F^*}(v) \leq_S \lambda_{G\setminus F}(v)$ and $\lambda_{G\setminus F^*}(w) \leq_S \lambda_{G\setminus F}(w)$. Therefore, $\lambda_{G\setminus F^*}(v)$ is incomparable to $\lambda_{G\setminus F^*}(w)$. It follows that if $u$ is not a transfer vertex in $C^*$, then $\lambda_{G\setminus F^*}(u) = \lambda_{G\setminus F}(u)$, and by repeated application of Lemma 13c to the ancestors of $u$, we have that $\lambda_{G\setminus F^*}(p) = \lambda_{G\setminus F}(p)$, so that $p$ is not a transfer vertex in $C^*$ by Lemma 13b.

(a) Assume $u \in D^*$. By the discussion above, $p$ is not a transfer vertex in $C^*$. If $p$ is unmarked in $C^*$, then $p$ is a d-move in $C^*$, which is a contradiction. So $p \in D^*$. But then, since $C^*$ has no moves, $(D^* \setminus \{u\}, F^*)$ has no moves and is therefore final contradicting the optimality of $C^*$. Hence, $u \notin D^*$.

(b) Assume $u$ is not a transfer vertex in $C^*$. Then $p$ is not a transfer vertex in $C^*$. If $p$ is unmarked in $C^*$, then $p$ is a d-move in $C^*$ which is a contradiction. Hence, $p \in D^*$.

■

*Lemma 16:* Only candidates are inserted into $Q$ in Algorithm 2.

*Proof:* Clearly, at the start of the first iteration of the while-loop on line 3, $Q$ contains only candidates.

Assume that $Q$ only contains candidates at the start of some iteration of the loop. Then, $(D, F)$ dequeued on line 4 is a candidate. Clearly, for any unmarked vertex $u$, $(D \cup \{u\}, F)$ is a candidate. Now, let $u$ be an s-move with children $v, w$. Let $F' = F \cup \{(u, w)\}$. We will now show that $F'$ is a transfer set. Let $(u', u'')$ be an edge in $F'$ distinct from $(u, w)$. Since $F$ is a transfer set, $\lambda_{G\setminus F}(u')$ is incomparable to $\lambda_{G\setminus F}(u'')$, and by Lemma 13a, we have that $\lambda_{G\setminus F'}(u') \leq \lambda_{G\setminus F}(u')$ and $\lambda_{G\setminus F'}(u'') \leq \lambda_{G\setminus F}(u'')$. Therefore, $\lambda_{G\setminus F'}(u')$ is incomparable to $\lambda_{G\setminus F'}(u'')$. Now consider the edge $(u, w)$. By the definition of an s-move, $\lambda_{G\setminus F}(w)$ is incomparable to $\lambda_{G\setminus F}(v)$, so that $\lambda_{G\setminus F'}(w)$ is incomparable to $\lambda_{G\setminus F'}(v)$. Since $\lambda_{G\setminus F'}(u) = \lambda_{G\setminus F'}(v)$, we see that $\lambda_{G\setminus F'}(u)$ is incomparable to $\lambda_{G\setminus F'}(w)$. Hence, $F'$ is a transfer set and $(D, F')$ is a candidate.

We can now see that, in all cases, only candidates are inserted into $Q$ during the while-loop.

■

All that remains is for us to show that the search we are performing really will find all optimal candidates. The proof of the next Lemma contains a detailed argument that shows that the order in which we consider the moves is not important.

*Lemma 17:* Each optimal final candidate for $S$, $G$, and $\sigma$ will be inserted into $Q$ at some point during the execution of Algorithm 2.

*Proof:* Let $C^*$ be an optimal candidate for $S$, $G$, and $\sigma$. If $|C^*| = 0$, then $C^*$ is inserted into $Q$ on line 2. Assume that $|C^*| > 0$. Consider one execution of Algorithm 2 and define a sequence of candidates $C_0, C_1, \ldots, C_m$, with $C_i = (D_i, F_i)$, as follows:

$$C_0 = (\emptyset, \emptyset).$$

If $C_i$ contains an s-move, then let $u$ with children $v, w$ be the s-move chosen on line 6 and define $C_{i+1}$ as:

$$C_{i+1} = \begin{cases} (D_i, F_i \cup \{(u,v)\}) & \text{if } (u,v) \in F^*, \\ (D_i, F_i \cup \{(u,w)\}) & \text{if } (u,w) \in F^*, \\ (D_i \cup \{p_{G \setminus F}(u)\}) & \text{otherwise.} \end{cases}$$

From Lemma 15, it is clear that $C_0, \ldots, C_m$ are all well defined, $C_0 \preceq C_1 \preceq \cdots \preceq C_m \preceq C^*$, and that $C_i$ contains an s-move for $i = 0, 1, \ldots, m-1$. Also, it is clear that $C_0, \ldots, C_m$ will all be inserted into $Q$ at some point during the execution of Algorithm 2.

We will now show that $F_m = F^*$. We already know that $F_m \subseteq F^*$. It remains for us to show that $F^* \subseteq F_m$. Consider the set $\beta$ of transfer vertices in $C^*$ that are not transfer vertices in $C$, i.e., $\beta = \{u : (u,v) \in F^* \setminus F_m\}$. Assume $\beta \neq \emptyset$. By Lemma 13b and d, $\beta$ contains an anchor w.r.t. $C_m$. Now, let $u \in \beta$ be an anchor w.r.t. $C_m$ that is $<_S$-maximally placed in $C_m$, by which we mean that if $u' \in \beta$ is an anchor w.r.t. $C_m$, then $\lambda_{G \setminus F_m}(u')$ is not a proper ancestor of $\lambda_{G \setminus F_m}(u)$. Let $v, w$ be the children of $u$ and assume, without loss of generality, that $(u,v) \in F^*$.

*Case 1.* $u$ is a root in the forest $G \setminus F_m$. $C^*$ contains no moves, and clearly, neither will the candidate $(D^*, F^* \setminus \{(u,v)\})$ contradicting the optimality of $C^*$.

*Case 2.* $u$ is not a root in $G \setminus F_m$ and $\lambda_{G \setminus F_m}(p_{G \setminus F_m}(u)) = \lambda_{G \setminus F_m}(u)$. Since $C_m$ contains no s-moves, $p_{G \setminus F_m}(u) \in D_m$. By the definition of $C_0, \ldots, C_m$, there is a $C_i$, $i < m$, such that $u$ is an s-move in $C_i$ and $p_{G \setminus F_i}(u) = p_{G \setminus F_m}(u)$. Again, by definition, $(u,v) \notin F^*$, which is a contradiction.

*Case 3.* $u$ is not a root in $G \setminus F_m$ and $\lambda_{G \setminus F_m}(p_{G \setminus F_m}(u)) \neq \lambda_{G \setminus F_m}(u)$. Let $p = p_{G \setminus F_m}(u)$. Clearly, $\lambda_{G \setminus F_m}(p)$ is a proper ancestor of $\lambda_{G \setminus F_m}(u)$. By the definition of $u$, any anchor $u'$ w.r.t. $C_m$ such that $\lambda_{G \setminus F_m}(u') = \lambda_{G \setminus F_m}(p)$ is not a transfer vertex in $C^*$. By Lemma 13d, $\lambda_{G \setminus F^*}(p) = \lambda_{G \setminus F_m}(p)$. It is now clear that since $C^*$ contains no moves, then $(D^*, F^* \setminus (u,v))$ contains no moves, contradicting the optimality of $C^*$.

In all cases, our assumption that $\beta \neq \emptyset$ leads to a contradiction. Hence, $F_m = F^*$.

If $D_m = D^*$, then $C_m = C^*$ and we are done. Assume instead that $D_m \subset D^*$. Define $C_{m+i}$ for $i = 1, \ldots, n$ as follows. If $C_{m+i}$ contains a d-move, then let $u$ be the d-move chosen on line 12 and define $C_{m+i+1}$ as

$$C_{m+i+1} = (D_{m+i} \cup \{u\}, F_{m+i}).$$

If $C_{m+i}$ contains no d-move, then $i = n$.

Clearly, $C_m \prec C^*$. Assume that $C_{m+i} \prec C^*$. Since $C_{m+i}$ contains no s-moves and is not final, it contains a d-move $u$. If $u \notin D^*$, then $u$ is a d-move of $C^*$. Therefore, $u \in D^*$. Hence, $C_{m+i+1} \preceq C^*$. By induction, $C_{m+n} \preceq C^*$. But $C_{m+n}$ contains no s-move or d-move and is therefore final. Hence, $C_{m+n} = C^*$. ■

Finally, we prove the time complexity of the algorithm. Note that the proof contains detailed descriptions of how the search for s-moves and d-moves can be implemented to achieve the time complexity.

*Theorem 3:* Given $S$, $G$, and $\sigma$, Algorithm 2 returns the set of optimal candidates and can be implemented to run in time $O(m + n \cdot 3^c)$, where $m = |V(S)|$, $n = |V(G)|$, and $c$ is the minimum cost of any final candidate for $S$, $G$, and $\sigma$.

*Proof:* The correctness of the algorithm is an easy consequence of Lemma 16 and Lemma 17. We will now proceed to show how the algorithm can be implemented to run in time $O(m + n \cdot 3^c)$.

By doing a one-time precomputation in time $O(m)$, it is possible to compute $\mathrm{lca}\,\{x, y\}$ for any pair of vertices $x, y \in V(S)$ in time $O(1)$. A clear exposition of how this can be done can be found in [37].

The while-loop of the algorithm is executed at most $O(3^c)$ times. To see this, note that each candidate $C$ that is dequeued on line 4 is replaced by at most three other candidates, each having a cost of $|C| + 1$. No candidate with cost exceeding $c + 1$ is ever inserted into $Q$. Hence the while-loop is executed at most $3^{c+1}$ times.

It only remains for us to show that every operation within the while-loop can be performed in time $O(n)$. More specifically, we have to show how to find an s-move or d-move in time $O(n)$.

We will assume that we can determine whether an arbitrary gene tree vertex $u$ is unmarked in constant time. First, we will precompute $\lambda_{G \setminus F}(u)$ for all vertices of $G$. Using the recursion in Lemma 2, we see that this can be done in time $O(n)$.

Next we precompute $p_{G \setminus F}(u)$. This can also be done in time $O(n)$ using the following recursion:

$$P[u] = \begin{cases} \emptyset & \text{if } u = \text{root}(G) \text{ or } (p_G(u), u) \in F, \\ P[p_G(u)] & \text{if } (p_G(u), u') \in F,\ u' \text{ sibling of } u, \\ p_G(u) & \text{otherwise.} \end{cases}$$

Clearly, if $u \in G \setminus F$, then $P[u] = p_{G \setminus F}(u)$.

Now, a gene tree vertex $u$ with children $v, w$ is an s-move iff $u$ is unmarked, $p_{G \setminus F}(u)$ is unmarked, $\lambda_{G \setminus F}(u) \neq \lambda_{G \setminus F}(v)$, $\lambda_{G \setminus F}(u) \neq \lambda_{G \setminus F}(w)$, and $\lambda_{G \setminus F}(p_{G \setminus F}(u)) = \lambda_{G \setminus F}(u)$. To check these conditions for each internal gene tree vertex takes time $O(n)$.

To find a d-move we simply check, for each vertex $u \in D \cup L(G)$, whether $p_{G \setminus F}(u)$ is unmarked and $\lambda_{G \setminus F}(p_{G \setminus F}(u)) = \lambda_{G \setminus F}(u)$. If so, $p_{G \setminus F}(u)$ is a d-move. Clearly, this can also be done in time $O(n)$.

Hence, Algorithm 2 can be implemented to run in time $O(m + n \cdot 3^c)$. ■

## VIII. A Note on Cycles and Gene Losses

As mentioned earlier, the interpretation of a transfer edge $(u, v)$ in a scenario is that a lateral transfer has occurred from the incoming edge of $\gamma(u)$ to some ancestral edge of $\gamma(v)$ that is not ancestral to $\gamma(u)$. So, although the starting point of each transfer in the species tree is made explicit in a scenario, the end point is not; the end point could be anywhere between $\gamma(v)$ and $\text{lca}\{\gamma(v), \gamma(u)\}$. This is in contrast to earlier work [32], where both the start and end points of lateral transfer events were made explicit. As such, the notion of cyclic scenarios is somewhat different.

In [32], a path was defined as a sequence $x_1, x_2, \ldots, x_n$ of species tree vertices where, for each pair of consecutive vertices $x_i$ and $x_{i+1}$, either $(x_i, x_{i+1})$ is an edge of the species tree, or $x_i$ and $x_{i+1}$ represent the edges between which a lateral transfer event has occurred. In the latter case, the transfer could be in any direction, i.e., either from the incoming edge of $x_i$ to the incoming edge of $x_{i+1}$, or vice versa. In this way, whenever we move from one vertex to another in a path we never move backward in time. A cycle is a path that starts and ends in the same vertex, and if a scenario contains a cycle, then the scenario is biologically infeasible. With our new definitions, a cyclic DTL-scenario would contain a cycle no matter where the end points of lateral transfer events were placed. We note here, that as pointed out in [32], cycles seem not to be a problem in practice.

Note, however, that the issue of transfer end points is the only essential difference between the old and new definitions. In fact, for each DTL-scenario we can find a corresponding set of old-style scenarios by explicitly choosing all possible end points for each transfer in the DTL-scenario. Hence, the set of most parsimonious scenarios are essentially the same irrespective of which definition we use.

We mentioned in section IV that there are many ways a gene tree could be mapped into a species tree, even when keeping the set of events, i.e., the sets $\Sigma$, $\Delta$, and $\Xi$, fixed. In our present context, we do not view the exact placement of the gene tree vertices within the species tree as an important question. Such questions are better answered by, for example, using sequence data. However, a more interesting question is whether or not every scenario with a fixed set of events is cyclic. In terms of the definitions in section VII, the question can be phrased: Given a final candidate $(D, F)$ for $S$, $G$, and $\sigma$, is there an acyclic scenario for $S$, $G$, and $\sigma$ such that $\Delta = D$ and $\Xi = F$?

The name of DTL-scenarios comes from the fact that we are using three biological events—duplications, lateral transfers, and gene losses—to explain the differences between species trees and gene trees. In the algorithms we have presented, however, we optimize the number of duplications and transfers only. There are several reasons for this. First, gene loss occurs frequently during evolution of genomes, where many duplications are followed by gene loss. Secondly, minimizing the number of losses can lead to the introduction of unnecessary and artificial transfer events. A duplication near the root of the species tree, for example, can lead to many losses further down the tree. However, a transfer could instead move the gene tree bifurcation closer to the leaves of the species tree, thereby eliminating losses. Instead, we propose to use the number of losses as a second criteria to choose between the different most parsimonious scenarios. A conservative approach for the detection of lateral transfer events would be to choose among the most parsimonious scenarios the ones with the fewest transfers, and among these, the ones with the fewest number of losses.

## IX. Empirical Performance

In this section we will analyze a biological dataset using the algorithms described previously. We will see how to handle some of the difficulties that may arise in dealing with real data, such as dealing with unrooted gene trees, and how to overcome them. Our results are comparable to other analyses done on the same data set, but we have the added advantage of being able to also take into account the role of duplications.

In [38], Matte-Tailliez *et al.* constructed phylogenies for 14 archaeal species: one based on the concatenation of 53 ribosomal proteins (7175 positions), which we will call $S_{\text{RP}}$, and one based on the concatenation of SSU and LSU rRNA (3933 positions), which we will call $S_{\text{rRNA}}$. These two species trees are shown in Fig. 6. In the same article, the authors also analyzed the impact of LGTs in their dataset and concluded that 8 genes may have undergone lateral gene transfer events. One of these is the *rpl12e* ribosomal protein, which we will study in this section. The same dataset was also analyzed by Jin *et al.* in [39].

The aligned rpl12e sequences were generously provided by Hervé Philippe. We used MrBayes [40] to obtain the ML gene tree shown in Fig. 7 (the tree with maximum posterior likelihood and the consensus tree were the same), which is identical to the one presented in [38] and has high edge posterior probabilities. We were not able to find any outgroup sequences that aligned well with the archaeal sequences, so instead of using an outgroup, we rooted the gene tree in all possible ways, thereby obtaining 25 candidate rooted gene tree $G_1, \ldots, G_{25}$.

Reconciling any of the rooted gene trees with $S_{\text{RP}}$ without using transfers, i.e., according to the duplication loss model, requires at least 7 duplications and 27 losses. For $S_{\text{rRNA}}$, the numbers are 6 duplications and 25 losses. We analyzed each pair of rooted gene tree and species tree using our algorithms. Note that the gene tree in Fig. 7 nicely groups the Crenarchaeota (*S. solfataricus, A. pernix,* and *P. aerophilum*), but internally, this clade is in conflict with each of the species trees in Fig. 6. This conflict can be reconciled using either one duplication and 3 losses or just one transfer. We will take the conservative approach here and use a duplication to explain the difference. A similar remark applies to the *Pyrococcus* clade which is different from that of $S_{\text{RP}}$.

We examined each of the scenarios obtained and discarded those that were considered highly unlikely and those with transfers within the Crenarchaeota or the *Pyrococcus* clade; For $S_{\text{RP}}$ at least 5 events, i.e., transfers or duplications, were needed. Among the cases with $d$ duplications and $t$ transfers, satisfying $d + t = 5$, we kept only those with a minimum number of losses. One gene tree whose root was placed in a very unlikely position, inside the *Pyrococcus* clade, was discarded. A similar analysis was performed for $S_{\text{rRNA}}$ where a minimum of 4 events were required. One scenario with 4 LGTs and no duplications turned out to be cyclic and was discarded. The results of the undiscarded scenarios for both species trees are summarized in Table I. The roots of the gene trees in Table I are highlighted in Fig. 7.

TABLE I

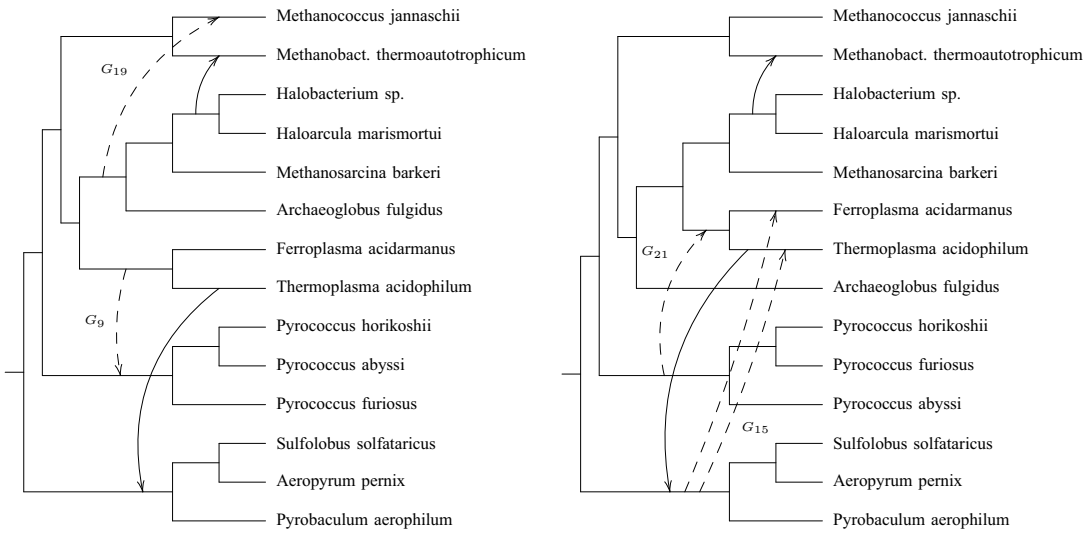
SUMMARY OF THE DTL-SCENARIOS RECONCILING THE ROOTED GENE TREES WITH THE TWO SPECIES TREES. THE COLUMNS D, T, AND, L CORRESPOND TO THE NUMBER OF DUPLICATIONS, TRANSFERS, AND LOSSES RESPECTIVELY.

| $S_{\text{RP}}$ | | | | $S_{\text{rRNA}}$ | | | |
|---|---|---|---|---|---|---|---|
| Gene Tree | D | T | L | Gene Tree | D | T | L |
| $G_{17}$ | 4 | 1 | 16 | $G_{17}$ | 3 | 1 | 15 |
| $G_{19}$ | 3 | 2 | 10 | $G_{19}$ | 2 | 2 | 8 |
| $G_9$ | 2 | 3 | 7 | $G_{15}$ | 1 | 3 | 5 |
| $G_{19}$ | 2 | 3 | 7 | $G_{21}$ | 1 | 3 | 5 |

For each pair of gene tree and species tree in Table I, we examined all most parsimonious scenarios and looked for common features. For $S_{\text{RP}}$, every most parsimonious scenario has a transfer to *Methanobacter thermoautotrophicum*, and every scenario except $G_{17}$ that has only one transfer, has also a transfer to the Crenachaeota. These are indicated with solid arrows in Fig. 6a. A scenario for $G_{19}$ has a transfer from the parent of *Archaeoglobus fulgidus* to *Methanococcus janaschii*, and a scenario for $G_9$ has a transfer from the parent of *Thermoplasma acidophilum* to the *Pyrococcus* clade. The latter two transfers are indicated with dashed arrows in Fig. 6a.

All scenarios for $S_{\text{rRNA}}$ has a transfer to *Methanobacter thermoautotrophicum*, just as for $S_{\text{RP}}$, and all scenarios except for $G_{15}$ and $G_{17}$ (the latter has only one transfer) has a transfer to the Crenarchaeota. The scenario for $G_{15}$ has instead two transfers in the opposite direction: from the least common ancestor of the Crenarchaeota to *Thermoplasma acidophilum* and to *Ferroplasma acidarmanus*. The scenario for $G_{21}$ has a transfer from the *Pyrococcus* clade to the parent of *Thermoplasma acidophilum*. See Fig. 6b.

There are similarities between our analysis of the data and that in [39]: e.g., there is a transfer to *M. thermautotrophicum*, and also a transfer from the Crenarchaeota to the least common ancestor of *T. acidophilium* and *F. acidarmanus* (similar to our scenario for $G_{15}$). A transfer within the *Pyrococcus* clade is present in [39], but as discussed above, we chose not to examine it further based on the fact that it may be just as easily explained by one duplication. The method given in [39] cannot handle duplications, and so uses transfers to explain any incongruency between the gene evolution model, the species tree, and the sequences. Finally, a transfer is indicated in [39] from the least common ancestor of *T. acidophilium* and *F. acidarmanus* to the Crenarchaeota *A. pernix*. No such transfer is




(a) $S_{\text{RP}}$ (b) $S_{\text{rRNA}}$

Fig. 6. The two organismal phylogenies from [38]. Fig 6a is based on 53 ribosomal proteins and 6b is based on SSU and LSU rRNA. See main text for the explanation of the transfer edges indicated in the figure.

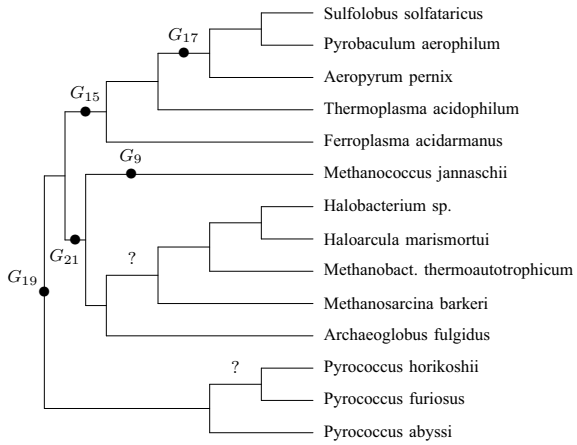



Fig. 7. The gene tree obtained from the archaeal rpl12e sequences. Note that the gene tree is unrooted. The highlighted positions show the roots of the indicated rooted gene trees. The edge posterior probabilities are all above 0.9 except for two edges that are indicated by question marks; the posterior probability of the edge in the *Pyrococcus* clade was 0.69, and for the other edge it was 0.86.

present in our analysis, and in fact, such a transfer contradicts the gene tree in Fig. 7.

Overall, there are significant similarities between the two analyses, although some transfers present in [39] clearly contradict the gene tree. For example, the scenario in [39] groups *A. pernix* together with *T. acidophilum* and *F. acidarmanus*. The posterior probability of such a clade (in the output from MrBayes) is close to zero. Our method also benefits from being able to consider duplications and transfers simultaneously.

## X. Open Problems and Discussion

In this paper, we have given a sound and biologically relevant definition of reconciliations between gene trees and species trees and devised algorithms for detecting most parsimonious reconciliations. For reasons that we explained in section VIII, we do not attempt to minimize the number of losses. But as we showed in our empirical tests, the number of losses can be used to choose among the most parsimonious scenarios when more than one exist. Moreover, we have seen that it can be of great use when the root of the gene tree under consideration is hard to determine.

The FPT-algorithm has a potential to be expanded in future work. First, if the minimum cost of reconciling $G$ and $S$ is known, then the algorithm can be easily modified to do a depth-first instead of a breadth-first search, something that will minimize the amount of memory needed. Second, it may be possible to extend the algorithm to search beyond the optimal scenarios and thereby provide the user the ability to search for non-parsimonious solutions if the most parsimonious ones are not satisfactory in light of other biological data.

It is also interesting to consider weighting duplications and LGTs differently. Ideally, such a weighting should reflect the likelihood of each event under a probabilistic model of evolution. An interesting question is whether there is an efficient algorithm for computing the optimal number of duplications and transfers for all weighting schemes simultaneously.

## References

[1] M. Lynch and A. Force, "The Probability of Duplicate Gene Preservation by Subfunctionalization," *Genetics*, vol. 154, no. 1, pp. 459–473, 2000.

[2] A. Force, M. Lynch, F. Pickett, A. Amores, Y. Yan, and J. Postlethwait, "Preservation of duplicate genes by complementary, degenerative mutations," *Genetics*, vol. 151, no. 4, pp. 1531–1545, Apr 1999.

[3] M. Lynch and J. Conery, "The evolutionary fate and consequences of duplicate genes," *Science*, vol. 290, no. 5494, pp. 1151–1155, Nov 2000.

[4] ——, "The evolutionary demography of duplicate genes," *J Struct Funct Genomics*, vol. 3, no. 1-4, pp. 35–44, 2003.

[5] M. Hahn, T. De Bie, J. Stajich, C. Nguyen, and N. Cristianini, "Estimating the tempo and mode of gene family evolution from comparative genomic data," *Genome Res*, vol. 15, no. 8, pp. 1153–1160, Aug 2005.

[6] J. Demuth, T. De Bie, J. Stajich, N. Cristianini, and M. Hahn, "The evolution of mammalian gene families," *PLoS ONE*, vol. 1, p. e85, 2006.

[7] J. Cotton and R. Page, "Rates and patterns of gene duplication and loss in the human genome," *Proc Biol Sci*, vol. 272, no. 1560, pp. 277–283, Feb 2005.

[8] J. Lawrence, "Horizontal and vertical gene transfer: the life history of pathogens," *Contrib Microbiol*, vol. 12, pp. 255–271, 2005.

[9] W. Doolittle and E. Bapteste, "Pattern pluralism and the tree of life hypothesis," *Proc Natl Acad Sci U S A*, vol. 104, no. 7, pp. 2043–2049, Feb 2007.

[10] M. Hallett and J. Lagergren, "Efficient algorithms for lateral gene transfer problems," *Proceedings of the Fifth Annual International Conference on Research in Computational Biology, April 22-25, 2001, Montreal, Canada*, 2001.

[11] E. Bapteste, E. Susko, J. Leigh, D. MacLeod, R. Charlebois, and W. Doolittle, "Do orthologous gene phylogenies really support tree-thinking?" *BMC Evol Biol*, vol. 5, no. 1, p. 33, 2005.

[12] M. Charleston, "Jungles: a new solution to the host/parasite phylogeny reconciliation problem," *Math Biosci*, vol. 149, no. 2, pp. 191–223, May 1998.

[13] L. Nakhleh, D. Ruths, and L. Wang, "Riata-hgt: A fast and accurate heuristic for reconstructing horizontal gene transfer," *Proceedings of the Eleventh International Computing and Combinatorics Conference (COCOON 05)*, pp. 84–93, 2005.

[14] V. Makarenkov and P. Legendre, "From a phylogenetic tree to a reticulated network," *J Comput Biol*, vol. 11, no. 1, pp. 195–212, 2004.

[15] E. Lerat, V. Daubin, H. Ochman, and N. Moran, "Evolutionary origins of genomic repertoires in bacteria," *PLoS Biol*, vol. 3, no. 5, p. e130, May 2005.

[16] D. Huson, *Split Networks and Reticulate Networks*. Oxford University Press, 2007, pp. 247–276.

[17] R. Azad and J. Lawrence, "Detecting laterally transferred genes: use of entropic clustering methods and genome position," *Nucleic Acids Res*, vol. 35, no. 14, pp. 4629–4639, 2007.

[18] M. Goodman, J. Czelusniak, G. Moore, A. Romero-Herrera, and G. Matsuda, "Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences," *Syst Zool*, vol. 28, no. 2, pp. 132–163, 1979.

[19] R. Guigó, I. Muchnik, and T. Smith, "Reconstruction of ancient molecular phylogeny," *Mol Phylogenet Evol*, vol. 6, no. 2, pp. 189–213, Oct 1996.

[20] B. Ma, M. Li, and L. Zhang, "From gene trees to species trees," *SIAM Journal on Computing*, vol. 30, no. 3, pp. 729–752, 2000.

[21] M. Hallett and J. Lagergren, "New algorithms for the duplication-loss model," *Proceedings of the fourth annual international conference on Research in Computational Molecular Biology*, pp. 138–146, 2000.

[22] I. Wapinski, A. Pfeffer, N. Friedman, and A. Regev, "Natural history and evolutionary principles of gene duplication in fungi," *Nature*, vol. 449, no. 7158, pp. 54–61, Sep 2007.

[23] L. Arvestad, A. Berglund, J. Lagergren, and B. Sennblad, "Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution," *Proceedings of the eighth annual international conference on Research in Computational Molecular Biology*, pp. 326–335, 2004.

[24] L. Arvestad, J. Lagergren, and B. Sennblad, "The gene evolution model and computing its associated probabilities," *J ACM*, vol. 56, no. 2, pp. 1–44, 2009.

[25] B. Sennblad and J. Lagergren, "Probabilistic orthology analysis," *submitted*, 2008.

[26] Ö. Åkerborg, B. Sennblad, L. Arvestad, and J. Lagergren, "Simultaneous bayesian gene tree reconstruction and reconciliation analysis," *Proc Natl Acad Sci U S A*, vol. 106, no. 14, pp. 5714–5719, Apr 2009.

[27] J. Lawrence and H. Ochman, "Molecular archaeology of the Escherichia coli genome," *Proc Natl Acad Sci U S A*, vol. 95, no. 16, pp. 9413–9417, Aug 1998.

[28] D. Gevers, K. Vandepoele, C. Simillon, and Y. Van de Peer, "Gene duplication and biased functional retention of paralogs in bacterial genomes," *Trends Microbiol*, vol. 12, no. 4, pp. 148–154, Apr 2004.

[29] A. Retchless and J. Lawrence, "Temporal fragmentation of speciation in bacteria," *Science*, vol. 317, no. 5841, pp. 1093–1096, Aug 2007.

[30] M. Csűrös and I. Miklós, "A probabilistic model for gene content evolution with duplication, loss and horizontal transfer," in *In Tenth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*. Springer, 2006, pp. 206–220.

[31] P. Górecki, "Reconciliation problems for duplication, loss and horizontal gene transfer," in *RECOMB '04: Proceedings of the eighth annual international conference on Resaerch in computational molecular biology*. New York, NY, USA: ACM, 2004, pp. 316–325.

[32] M. Hallett, J. Lagergren, and A. Tofigh, "Simultaneous identification of duplications and lateral transfers," *Proceedings of the eighth annual international conference on Research in Computational Molecular Biology*, pp. 347–356, 2004.

[33] L. Addario-Berry, M. Hallett, and J. Lagergren, "Towards identifying lateral gene transfer events," *Proc 8th Pacific Symp on Biocomputing (PSB03)*, pp. 279–290, 2003.

[34] R. D. M. Page, "Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas," *Systematic Biology*, vol. 43, no. 1, pp. 58–77, 1994.

[35] B. Mirkin, I. Muchnik, and T. Smith, "A biologically consistent model for comparing molecular phylogenies." *J Comput Biol*, vol. 2, no. 4, p. 493, 1995.

[36] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979.

[37] M. A. Bender, M. Farach-Colton, G. Pemmasani, S. Skiena, and P. Sumazin, "Lowest common ancestors in trees and directed acyclic graphs," *J Algorithms*, vol. 57, no. 2, pp. 75–94, 2005.

[38] O. Matte-Tailliez, C. Brochier, P. Forterre, and H. Philippe, "Archaeal phylogeny based on ribosomal proteins," *Mol Biol Evol*, vol. 19, no. 5, pp. 631–639, May 2002.

[39] G. Jin, L. Nakhleh, S. Snir, and T. Tuller, "Maximum likelihood of phylogenetic networks," *Bioinformatics*, vol. 22, no. 21, pp. 2604–2611, Nov 2006.

[40] F. Ronquist and J. Huelsenbeck, "MrBayes 3: Bayesian phylogenetic inference under mixed models," *Bioinformatics*, vol. 19, no. 12, pp. 1572–1574, Aug 2003.

# II

# Inferring Duplications and Lateral Gene Transfers— An Algorithm for Parametric Tree Reconciliation

Ali Tofigh and Jens Lagergren

**Abstract**

Prediction of the function of genes and their products is an increasingly important computational problem. The ability to correctly identify the historic relationship of homologous genes is essential for making accurate predictions. In 1970, Fitch made a distinction between paralogous and orthologous genes, its importance lying in the observation that genes are more likely to have similar functions when they have evolved from a common ancestral gene through speciation rather than duplication. Lateral gene transfer (LGT) is yet another important evolutionary event that creates copies of genes, and as our understanding of the importance and prevalence of LGT in evolution is deepening, there is a high demand for methods for detection of LGTs when reconstructing the evolutionary past of genes.

In this paper, we present highly efficient and practical algorithms for tree reconciliation that simultaneously consider both duplications and LGTs. We allow costs to be associated with duplications and LGTs and develop methods for finding reconciliations of minimal total cost between species trees and gene trees. Moreover, we provide an efficient algorithm for parametric tree reconciliation—a computational problem analogous to parametric sequence alignment. Experimental results on synthetic data indicate that our methods are robust with high specificity and sensitivity.

## 1 Introduction

Crucial to prediction of gene function is reconstruction of the historic relationship of homologous genes and detection of evolutionary events responsible for shaping the genomes of species. The fate of a single gene is determined by both large scale and small scale events. The nucleotide composition of a gene, and thereby the function of its products, is affected by events such as mutations, insertions, deletions, exon shuffling, exon duplications, and gene conversion. Genes are lost by chromosomal deletions or silencing mutations. New genes are born via events that duplicate a gene, e.g. via segmental duplication, or via other events such as interstitial deletions—deletions of internal regions of chromosomes that can

result in the formation of chimeric genes. Genes can also be transferred between different organisms via lateral gene transfers (LGTs), also known as horizontal gene transfers. Studying the evolution of genes and their history is an important step towards the prediction of the function of genes in genomes.

There is a long history of using trees to describe relationships among species as well as among sets of homologous genes from different organisms. It is quite common that reconstructed gene trees differ topologically from their corresponding organismal phylogenies. The evolutionary events mentioned in the previous paragraph as well as population genetic effects may be the cause of such incongruities. In this article, we will consider genes as atomic units that evolve inside an organismal phylogeny, and we will consider three evolutionary events when explaining the differences in topology between gene and species trees, namely gene duplication, gene loss, and lateral gene transfer.

Gene duplication has long been known to be a major factor driving the evolution of genomes [1, 2] and the rate with which gene duplication occurs in different parts of the tree of life has been extensively studied [3–7]. In [8, 9], the fates of recently duplicated genes were termed non-functionalization (loss of function), sub-functionalization (where each copy takes on a subset of the original function), and neo-functionalization (where one copy assumes a new function). In [3], the frequency of gene duplications among a set of eukaryotic species was estimated to be approximately the same as individual nucleotide substitutions: 0.01 gene duplications per gene per million years.

The realization that gene duplications and losses can create incongruities between a gene tree and a corresponding species tree has lead to the formulation of several interesting phylogenetic problems. Already in 1979, Goodman *et al.* gave a parsimony method in which a gene tree is embedded in the species tree such that the number of duplications and losses required to explain the gene evolution is minimized [10]. Guigó *et al.* continued this work by attempting to find the species tree that explains a set of gene trees with a minimum number of duplications [11]. Ma *et al.* proved hardness results for several variations of species tree reconstruction problems [12]. Due the intrinsic hardness of species tree reconstruction, several heuristics have been developed, e.g., [13–15]. Assuming that one of the gene trees has had a constant number of lineages in each species tree lineage, Hallett *et al.* gave an efficient algorithm for finding an optimal species tree [16].

Contrary to gene duplications, the importance and prevalence of lateral gene transfers have been the subject of much controversy. The possibility of lateral gene transfers in bacteria was realized already in 1946 [17,18] and demonstrated to occur between different bacterial species in 1959 [19]. Its occurrence among prokaryotes has since been well documented, see for example [20] and [21]. Evidence has also been presented for the occurrence of lateral gene transfers from prokaryotes to

eukaryotes and even between eukaryotes, see [22] for a recent review. Overall, the importance of HGT is today recognized as a major force of evolution, in particular among prokaryotes. In fact, due to its apparent impact on prokaryotic evolution, the appropriateness of using species trees to represent the evolutionary history of certain taxa have been questioned [23–25], see also [26] and references therein.

Here, we will adopt an intermediate view with respect to prokaryotic evolution that has emerged in recent years, namely that although LGT is common, it is not so common among the genes of a species that phylogenetic trees cannot meaningfully represent the history of organismal evolution [27]. When accepting this intermediate view, we are faced with relevant and important computational challenges. These include reconstructing species trees and gene trees, and to ask if and where LGT has occurred among homologous genes.

The parsimony version of phylogenetic detection of LGTs was formalized and treated in [28]. Other heuristics for the problem include [29–31].

There have been few attempts to devise phylogenetic methods for the simultaneous detection of duplications and lateral gene transfers. Early work in this direction was performed in the related field of host-parasite co-evolution [32]. Host and parasite phylogenies can differ for reasons similar to that of species trees and gene trees: speciation of a parasite independent of its host, host switching, and lineage sorting correspond to duplication, LGT, and gene loss, respectively. A similar comparison can be made to biogeography, where species track geographical areas much in the same way that parasites track hosts and genes track organisms, see [33] for an overview. Although the method developed in [32] considered both duplications and LGTs (in the context of host-parasite phylogenies) and was an excellent first attempt, the presentation in [32] is not mathematically sound and the time complexity of the algorithm is unclear. The first mathematically rigorous formalization appeared in [34, 35].

Methods that take sequence information into account directly, rather than via a gene tree, have started to emerge. For example, methods have been proposed for so-called duplication analysis where construction of a gene tree from sequences, the embedding of the gene tree within the corresponding species tree, and the identification of duplications are all considered simultaneously. An *ad hoc* method for duplication analysis that also considers gene order information was presented in [36]. A probabilistic model of gene evolution for the duplication-loss model and computational tools for duplication analysis have been developed in [37–40].

Atypical sequence information has been used in several cases to detect recent lateral gene transfers, for example [41]. Probabilistic methods have also been developed that utilize models of sequence evolution on a species tree to detect LGTs, for example [42]. In the context of hosts and parasites, Huelsenbeck *et al.* developed a Bayesian framework for the detection of host switches using Markov

chain Monte Carlo and taking advantage of sequence information from the host and parasite species [43]. The model in [43] assumes a one-to-one correspondence between hosts and parasites and does not consider duplications.

To our knowledge, the first probabilistic method for simultaneous analysis of duplications and LGTs was proposed just recently in [44], although a probabilistic model based on the birth-death process [45] was used in [34] to generate synthetic data, and a similar model was used in [46] to estimate gene family sizes,

In this article, we expand on previous work in [34] and [35] where a duplication-transfer-loss model of gene evolution was described in a parsimony setting. We improve the time complexity of our algorithms and extend our methods by allowing costs to be associated with the events. In this way, more refined analyses of duplications and LGTs may be performed and a greater degree of freedom is given to the user to make adjustments according to prior information about the relative prevalence of duplications and LGTs.

We also provide an algorithm for the parametric version of the tree reconciliation problem. This is analogous to parametric sequence alignment where regions of the parameter space are sought such that the set of optimal solutions for every point in the same region is identical [47–49]. We provide an efficient algorithm that partitions the space of duplication and LGT costs into regions in which the set of optimal solutions is identical for any set of points in the same region.

The outline of the paper is as follows. In section 3, the duplication-loss and duplication-transfer-loss models of gene evolution are thoroughly discussed. Section 4 provides the definition of DTL-scenarios and discusses various related issues such as gene losses and temporal feasibility of reconciliations. In section 5, we improve on previous algorithms for the parsimony version of tree reconciliation by providing a dynamic programming algorithm with a significantly lower time complexity that also allows costs to be associated with duplication and transfer events. Section 6 contains an algorithm for parametric tree reconciliation. Finally, in section 7, we provide empirical tests of our algorithms on different sets of synthetic data.

## 2 Definitions

In this article, we follow the same notational conventions as in [35]. Although most of the notation is standard in the field, we will comment on a few conventions used in this article.

We deal only with rooted binary trees. The edges of a tree $T$ are assumed to be directed away from the root. We write $v \leq_T u$ to denote that $v$ is a descendant of $u$ in the tree $T$. Note that each vertex is a descendant of itself. When $v$ is a descendant of $u$ distinct from $u$, we write $v <_T u$ and $v$ is said to be the proper

descendant of $u$. If $v$ is a (proper) descendant of $u$, we also say that $u$ is a (proper) ancestor of $v$.

# 3 Models of Gene Evolution and DTL-scenarios

In [35], we developed methods for reconciling the differences between a gene tree and a corresponding species tree while simultaneously considering both duplications and LGTs. There, we defined Duplication-Transfer-Loss scenarios (DTL-scenarios) to serve as our formal notion of reconciliations. Here, we expand on that work by providing new methods and improvements on those presented in [35]. First, we give a review of DTL-scenarios.

We assume that the data to be analyzed consists of a gene tree $G$, in the form of a rooted binary tree, and a corresponding species tree $S$, also a rooted binary tree. The correspondence between $S$ and $G$ is given via a function $\sigma : L(G) \to L(S)$ that maps each gene to the extant species to which it belongs. A DTL-scenario for $S$, $G$, and $\sigma$, as defined in the next section, gives a biologically feasible history of the evolution of $G$ inside $S$ by mapping the gene tree into the species tree and assigning to each internal gene tree vertex either a speciation, duplication, or LGT event. The conditions for a DTL-scenario ensure that the mapping of $G$ into $S$ and the assignment of events to the bifurcations of $G$ are biologically consistent. Also, we must pay attention to the direction of time which exists implicitly in the trees. Before giving the formal definition, we will explain some of the reasoning that went into its formulation. We begin with a short review of the duplication-loss model of gene evolution.

## 3.1 The Duplication-Loss Model

In the duplication-loss model of gene evolution, each bifurcation of a gene tree $G$ represents either a speciation or duplication that has occurred during the evolution of the genes inside a corresponding species tree $S$. Each gene tree vertex that represents a speciation is associated with a species tree vertex (the one that represents the same speciation event). Each gene tree vertex that represents a duplication is associated with a species tree edge (the edge along which the duplication occurred). It is clear that the true history of the evolution of the genes can be depicted as $G$ evolving strictly within the edges of $S$, see Figure 1 for an example.

Although duplications and speciations are made explicit in such a reconciliation, losses exist only implicitly. They can, however, be inferred by examining each gene tree edge and the path it takes in the species tree, see for example [50] for a formal treatment. Informally, if $(u, v)$ is a gene tree edge, then at least one loss is

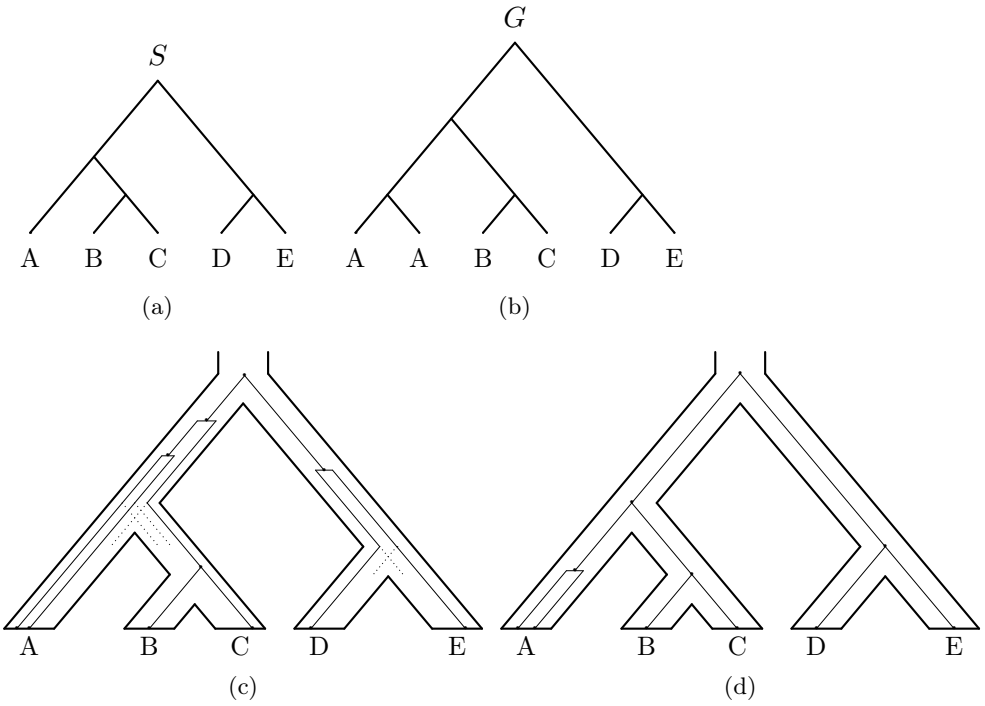



Figure 1: **Examples of how a gene family evolves inside a species tree according to the duplication-loss model.** A species tree and a corresponding gene tree are shown in (a) and (b), respectively. (c) is an example of a possible evolutionary history of the gene tree according to the duplication-loss model. Note that in this example, the history consists of three duplications and five losses. In (d), the gene tree has been mapped into the species tree using the least common ancestor mapping that minimizes the number of duplications and losses, which in this case are one and zero, respectively.

inferred for each species tree vertex that lies strictly between the path from $u$ to $v$ in $S$ (exact definitions are given later).

When treating a reconciliation mathematically, we need to formulate a precise definition for the association of gene tree vertices with vertices and edges of $S$. A convenient way to do this is to map the gene tree vertices to species tree vertices via a function that extends $\sigma$, say $f : V(G) \to V(S)$. The interpretation of this mapping is then as follows: If $u$ is a duplication in the gene tree, then the duplication has occurred along the incoming edge of $f(u)$. If $u$ is a speciation, then $f(u)$ is the speciation event that caused the bifurcation.

Given a gene tree and a species tree, we may ask what the true reconciliation is. To answer this computationally, we can define, based on biological studies or input from biologists, suitable cost functions on the set of possible mappings of $G$ into $S$ and search for those with optimal cost. Of course, we must also define exactly what constitutes a valid mapping. Necessary and sufficient conditions to ensure that a mapping $f : V(G) \to V(S)$ is biologically and temporally feasible is that it extends $\sigma$ and that for each gene tree edge $(u, v)$, $f(u) \geq_S f(v)$. This ensures that the mapping has a valid biological interpretation and can be depicted with figures such as those in Figure 1.

However, the sets of gene tree vertices corresponding to duplications and speciations are not made explicit by such a mapping. In fact, each such mapping corresponds to a, possibly large, set of different reconciliations. This disparity between mappings and reconciliations is due to the fact that a mapping does not make explicit which gene tree vertices are associated with species tree edges, and which are associated with species tree vertices. However, given a mapping, only certain gene tree vertices can feasibly be classified as speciations, namely those whose children are mapped to incomparable species tree vertices. The rest of the internal gene tree vertices must be classified as duplications. Given a mapping $f$, a vertex of $G$ could conceivable be a speciation only if its children are mapped to incomparable species tree vertices, say $x$ and $y$, and who is itself mapped to the least common ancestor of $x$ and $y$; all others must be duplications.

Most research dealing with reconciliation of trees has been performed in a parsimony setting where it turns out that there is a unique mapping of the gene tree into the species tree that simultaneously minimizes the number of inferred losses and duplications. We call this mapping the least common ancestor mapping and it is defined as:

$$\lambda(u) = \mathrm{lca}\,(\sigma(L(G_u))),$$

where we slightly abuse our notation by letting $\sigma$ map *sets* of gene tree leaves to the corresponding *sets* of species tree leaves. See Figure 1d for an example of using $\lambda$ to map a gene tree inside a species tree. Since $\lambda$ is the unique most parsimonious mapping, usually no other reconciliations are considered. Instead,

the focus has mainly been on finding the species tree that minimizes the total number of duplications and losses with respect to a set of gene trees. In [51], a more general definition of a mapping was given that explicitly mapped gene tree vertices to either species tree vertices or edges. There is also a need to deal with non-parsimonious reconciliations in probabilistic models of gene evolution, see for example [37].

### 3.2 The Duplication-Transfer-Loss Model

The picture becomes significantly more complex when we also consider lateral gene transfers. An LGT event involves one gene and two different but contemporary species, and can be depicted as an arc between two edges of a species tree. The evolution of genes in the duplication-loss model is fully contained within the edges of the species tree. For example, a single edge of the gene tree represents direct inheritance of the gene from generation to generation along a set of species tree edges. Hence, although there may have been intermediate duplications followed by loss along a gene tree edge, the edge still represents a path in the species tree. In the duplication-transfer-loss model (DTL-model), however, the history of a gene tree edge could be rather complex. A gene may for example be transferred to another species and later be lost in the species from which the transfer originated. This process could conceivably be repeated many times before a branching occurs such that both copies have surviving descendants at the present time. Hence, the true evolutionary history represented by a gene tree edge in the DTL-model may include several species tree edges, i.e., not necessarily a path in the species tree. Figure 2 shows a possible reconciliation between the same trees as in Figure 1. See Figure 3 for several examples of complicated and perhaps unlikely histories that are conceivable in a general DTL-model.

In a parsimony setting, not every possible reconciliation needs to be considered. Just as only one mapping, the least common ancestor mapping, is used in the duplication-loss model in a parsimony setting, we only wish to consider a relevant subset of all possible reconciliations in the DTL-model. For example, assume that both endpoints of a gene tree edge are mapped to comparable vertices or edges of the species tree in the true reconciliation. Even if the history of that edge contains transfer events, like the situation in Figure 3b, we cannot really hope to be able to reconstruct such a history (at least not only with data in the form of phylogenetic trees). Hence, in our DTL-scenarios, we will consider an edge as a transfer edge if and only if the endpoints are placed at incomparable locations within the species tree.

Classification of vertices introduces yet another complexity. It is conceivable that a bifurcation of the gene tree is due to a duplication or a speciation, while

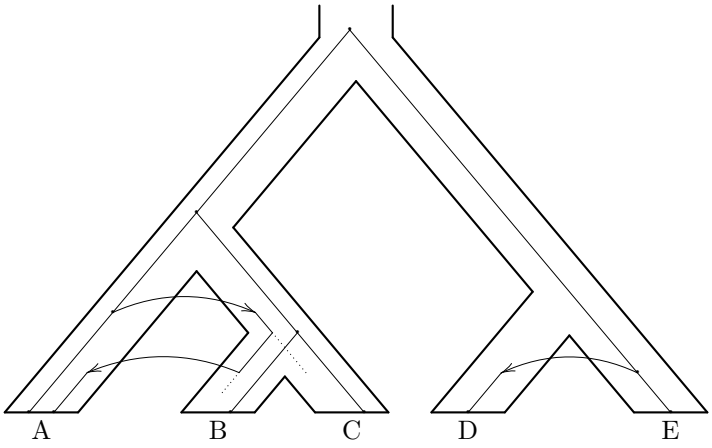



Figure 2: **An example of a gene family evolving inside a species tree according to the DTL-model.** Note how a gene is transferred from species A and back without leaving a trace in other species. There is really no hope of being able to recover such evolutionary histories, which leads to us instead classifying the bifurcation caused by the transfer as a duplication.

at the same time, one of the outgoing edges is a transfer edge. But for a transfer edge we cannot hope to gain insight from our data about the true cause of the bifurcation represented by the parent vertex. For example, we cannot distinguish between the simple case of lateral gene transfer and the case of a duplication followed by the transfer of one of the copies and the subsequent loss of the gene from which the transfer originated (Figure 3c). A similar remark can be made about a transfer to a second species and back. Since the placement of both parent and child are comparable in the species tree, we can only hope to classify the parent as a duplication. Hence, the approach we will take is to classify a gene tree vertex as a transfer vertex if and only if one of its outgoing edges is a transfer edge.

Finally, we must consider the case when both outgoing edges of a gene tree vertex are transfer edges (Figure 3e). Though such a case is biologically conceivable, we consider it as degenerate in a parsimony setting. Hence, we will also restrict our attention to those reconciliations where at most one of the outgoing edges of each gene tree vertex is a transfer edge.

Even with the above restrictions, when considering different mappings of the gene tree into the species tree, there is no single best mapping that minimizes the number of events, as was the case in the duplication-loss model. First, there is a trade-off between using duplications and transfers to explain incongruities between a gene tree and a species tree. Also, the consideration of time that is implicitly present in the trees is much more complicated in the DTL-model. For

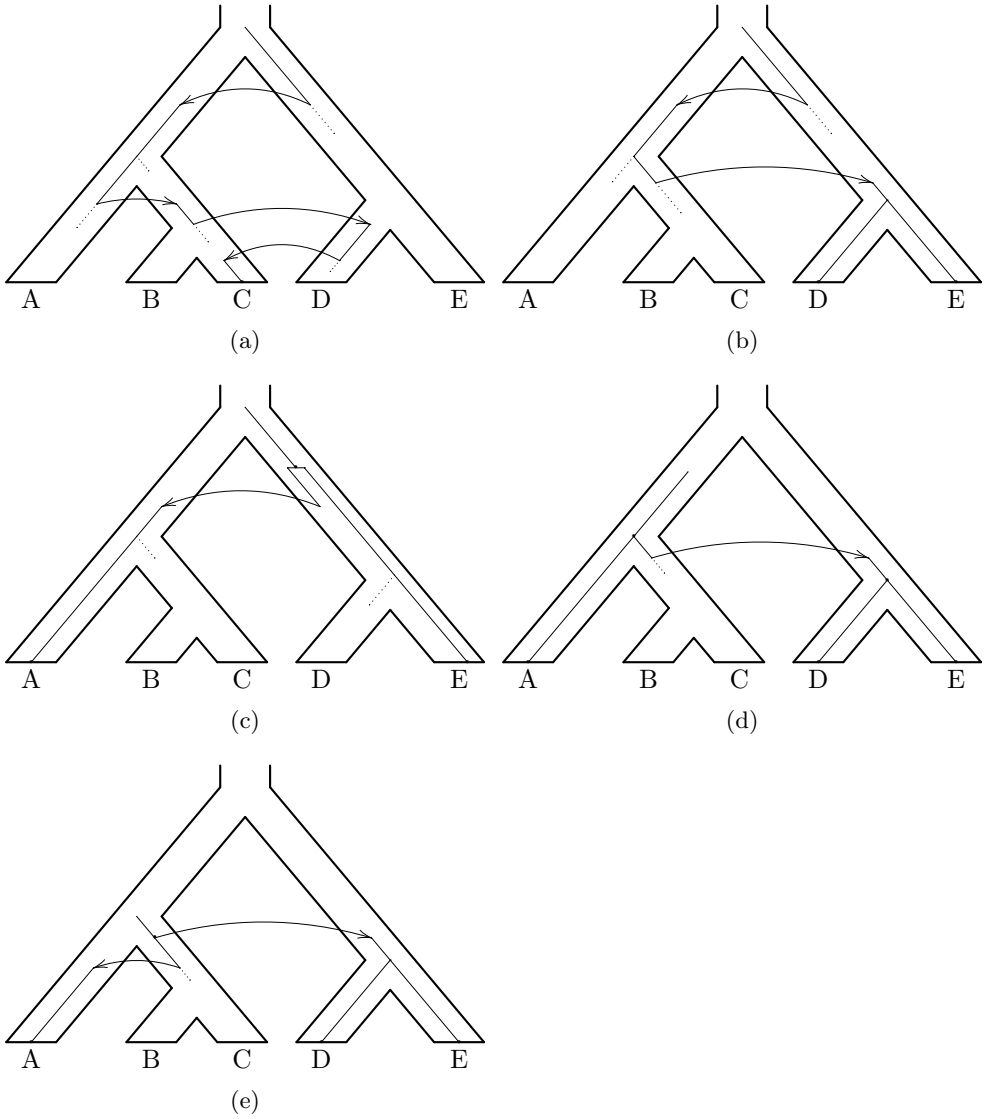



Figure 3: **Examples of complicated and unlikely evolutionary histories in the DTL-model.**

these reasons, we will need to allow a greater amount of freedom when defining valid mappings compared to the duplication-loss model, while discarding the degenerate cases discussed above. This then leads to the definition of DTL-scenarios as stated in the next section.

# 4 DTL-scenarios

The definition of DTL-scenarios presented here is taken from [35] where they were first defined. Informally, a DTL-scenario partitions the internal gene tree vertices into three parts corresponding to speciations, duplications, and transfers; these parts will be denoted $\Sigma$, $\Delta$, and $\Theta$, respectively. The gene tree is mapped into the species tree via a function $\gamma : V(G) \to V(S)$. The set of transfer edges, i.e., those gene tree edges whose endpoints are mapped by $\gamma$ to incomparable species tree vertices are denoted by $\Xi$. A transfer edge $(u, v)$ represents an LGT that has occurred from the incoming edge of $\gamma(u)$ to some edge along the path from $\text{lca}\,\{\gamma(u), \gamma(v)\}$ to $\gamma(v)$.

Formally, A DTL-scenario for a species tree $S$, a gene tree $G$, and a leaf-mapping function $\sigma : L(G) \to L(S)$ is an octuple

$$(S, G, \sigma, \gamma, \Sigma, \Delta, \Theta, \Xi),$$

where $S$ and $G$ are rooted binary trees, $\sigma : L(G) \to L(S)$ is a function, $\gamma : V(G) \to V(S)$ is an extension of $\sigma$, $\Sigma$, $\Delta$, and $\Theta$ form a partition of $\mathring{V}(G)$, and $\Xi \subset E(G)$ is a subset of the gene tree edges such that:

(I) If $u \in \mathring{V}(G)$ is a gene tree vertex with children $v$ and $w$, then

  (a) $\gamma(u)$ is not a proper descendant of $\gamma(v)$ or $\gamma(w)$
  (b) At least one of $\gamma(v)$ and $\gamma(w)$ is a descendant of $\gamma(u)$

(II) $(u, v) \in \Xi$ if and only if $\gamma(u)$ is incomparable to $\gamma(v)$

(III) If $u \in \mathring{V}(G)$ is a gene tree vertex with children $v$ and $w$, then

  (a) $u \in \Theta$ if and only if $(u, v) \in \Xi$ or $(u, w) \in \Xi$
  (b) $u \in \Sigma$ only if $\gamma(u) = \text{lca}\,\{\gamma(v), \gamma(w)\}$ and $\gamma(v)$ and $\gamma(w)$ are incomparable
  (c) $u \in \Delta$ only if $\gamma(u) \geq_S \text{lca}\,\{\gamma(v), \gamma(w)\}$

As mentioned in the previous section, our goal is to assign costs to reconciliations and find those that are optimal. In [35] the cost of a DTL-scenario was

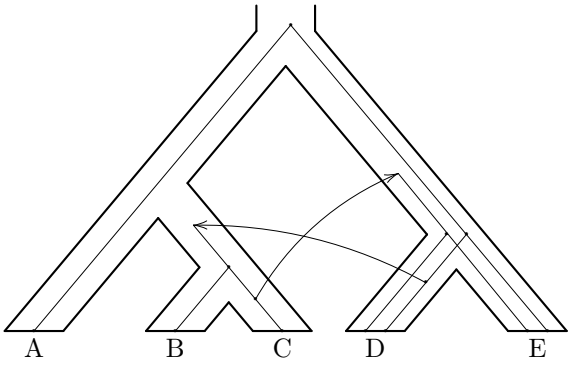



Figure 4: **An example of a cyclic DTL-scenario.** By Definition 1, the DTL-scenario in the figure is acyclic since there is no way to linearly order the species tree vertices such that the parent of species D comes before the parent of species B and vice versa.

simply the number of duplications and LGTs. Here we will allow different costs to be attributed to duplications and LGTs. If the duplication cost is $C_{\mathrm{d}}$ and the transfer cost is $C_{\mathrm{t}}$, then the cost of a DTL-scenario $\alpha$ is defined by

$$|\alpha| = |\Delta| \cdot C_{\mathrm{d}} + |\Theta| \cdot C_{\mathrm{t}} = |\Delta| \cdot C_{\mathrm{d}} + |\Xi| \cdot C_{\mathrm{t}}$$

A DTL-scenario is called optimal if and only if its cost is minimum among the set of all DTL-scenarios for $S$, $G$, and $\sigma$.

Following [35], we do not minimize the number of losses as this can lead to an overestimation of the number of LGTs. The approach we take here is the same as that in [35]; we propose that the number of losses should be used to refine the set of optimal DTL-scenarios. It is, however, not too difficult to extend our algorithms to also take losses into account.

## 4.1 Cycles

One of the difficulties in finding biologically feasible reconciliations in the DTL-model arises from the implicit notion of time present in the trees. As stated in section 3.1, this does not cause any problems in the duplication-loss model. DTL-scenarios that are temporally infeasible are called cyclic, an example of which is shown in Figure 4. In this section, we discuss the technical aspects of cycle detection and the difficulties that cycles create. But first, we formally define what constitutes a cyclic DTL-scenario.

In a DTL-scenario, the interpretation of a transfer edge $(u, v) \in \Xi$ is that a lateral gene transfer event has occurred between the incoming edge of $\gamma(u)$ to some ancestral edge of $\gamma(v)$. However, this edge cannot be ancestral to $\gamma(u)$ since that would imply a transfer from a species to one of its ancestors. Hence, as mentioned

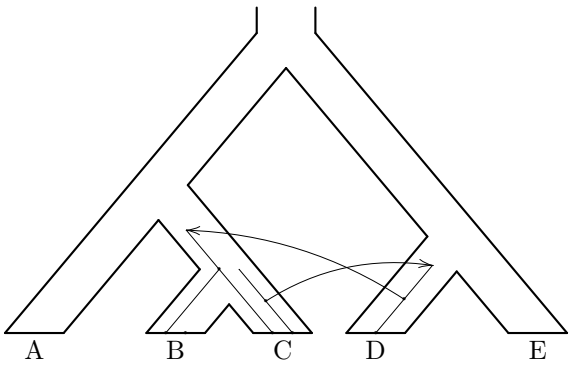



Figure 5: **Part of a DTL-scenario that is acyclic according to definition used in Jungles, but is temporally infeasible.** By the definition of acyclicity in Jungles, we only need to find a linear ordering of the species tree vertices such that the parent of species D comes before the parent of species C, and the parent of species C comes before species D. An example of such a linear order is the root, then the parent of A, the parent of D, the parent of C, and finally the leaves in any order. However, the DTL-scenario is clearly temporally infeasible.

in section 4, we interpret the transfer as having occurred between the incoming edge of $\gamma(u)$ and some edge along the path from $\text{lca}\{\gamma(u), \gamma(v)\}$ to $\gamma(v)$. In other words, a DTL-scenario makes explicit the location in the species tree from which a transfer has occurred but not the point to which the gene was transferred.

A similar idea was used in Jungles [32] where the "landing point" of a transfer was allowed to be moved closer to the root. However, although the condition used in Jungles to define temporally infeasible reconciliations is necessary, it is not sufficient. The main idea, recast in terms of our DTL-scenarios, can be described as follows. A reconciliation in [32] was considered temporally feasible if the partial order on the species tree vertices induced by the edges of the species tree could be extended to a linear order such that for each transfer edge $(u, v)$, the parent of $\gamma(u)$ appeared before $\gamma(v)$ in the linear order. This guarantees that for each transfer edge $(u, v)$, the incoming edge of $\gamma(u)$ is contemporary to some ancestral edge of $\gamma(v)$. Hence, a temporally appropriate landing point can always be chosen for each transfer edge.

The problem with this definition is that it does not fully consider the temporal aspect of the gene tree. Figure 5 shows a DTL-scenario that satisfies the above condition but is clearly temporally infeasible. Here we give the definition of acyclic DTL-scenarios that was used in [35] and prove that it is both necessary and sufficient to ensure temporal feasibility.

**Definition 1.** We say that a DTL-scenario is acyclic if and only if there is a total order $<$ on $V(S)$ such that

*a.* if $(x, y) \in E(S)$, then $x < y$,

*b.* if $(u, v), (u', v') \in \Xi$ and $v \geq_G v'$, then $p(\gamma(u)) < \gamma(v')$.

To prove that the above definition is both necessary and sufficient we prove its equivalence to the following alternative definition. This definition makes the notion of time explicit in both the species tree and the gene tree thus dispensing with any doubts concerning its necessity or sufficiency.

**Definition 2.** We say that a DTL-scenario is temporally feasible if and only if there exists a time function $t : V(S) \cup V(G) \to \mathbb{R}$ such that

*a.* $t|_{V(S)}$ is one-to-one,

*b.* if $(x, y) \in E(S)$, then $t(x) < t(y)$,

*c.* if $(u, v) \in E(G)$, then $t(u) < t(v)$,

*d.* if $u \in \Sigma \cup L(G)$, then $t(u) = t(\gamma(u))$,

*e.* if $u \in \Delta \cup \Theta$, then $t(p(\gamma(u))) < t(u) < t(\gamma(u))$ where $t(p(\text{root}(S))) = -\infty$.

**Theorem 1.** *A DTL-scenario is acyclic if and only if it is temporally feasible.*

*Proof.* Let $\alpha = (S, G, \sigma, \gamma, \Sigma, \Delta, \Theta, \Xi)$ be scenario for $S$, $G$, and $\sigma$.

Assume that $\alpha$ is temporally feasible, and let $t$ be a time function for $\alpha$ satisfying the conditions of Definition 2. Order the vertices of $S$ according to $t$, i.e., $x < y$ iff $t(x) < t(y)$. Clearly, $<$ is a total order on the vertices of $S$ and satisfies $1a$. Let $(u, v), (u', v') \in \Xi$ such that $v \geq_G v'$. By $2c$, we have that $t(u) < t(v')$. Hence, by $2d$ and $2e$, we have that

$$t(p(\gamma(u))) < t(u) < t(v') \leq t(\gamma(v')),$$

and by the definition of our order $<$ on $V(S)$, we see that $p(\gamma(u)) < \gamma(v')$, so that $1b$ is also satisfied.

To prove the other direction, assume that $\alpha$ is acyclic. Let $<$ be a total order on the vertices of $S$ satisfying $1a$ and $1b$. We will show how to define a time function $t : V(G) \cup V(S) \to \mathbb{R}$ that satisfies the conditions of definition 2.

Let $t$ assign, in an arbitrary fashion, distinct real numbers to each vertex of $S$ according to the order $<$ such that $t(x) < t(y)$ iff $x < y$. For $u \in \Sigma \cup L(G)$, let $t(u) = t(\gamma(u))$. Note that $2a$, $2b$, and $2d$ are satisfied. Next, we will assign times to the transfer vertices of $G$. We will do this recursively from the root towards the leaves (inorder). If $(u, v) \in \Xi$, then define

$$A(u) = \{\gamma(v') : v' \leq_G v, (u', v') \in \Xi\} \cup \{\gamma(u)\},$$

and

$$B(u) = \{u' : (u', v') \in \Xi, u' >_G u\} \cup \{p(\gamma(u))\}.$$

Note that by $1b$, we have that $p(\gamma(u)) < x$ for each $x \in A(u)$, which in turn implies $t(p(\gamma(u))) < t(x)$. Thus, assuming that each gene tree vertex $u' \in B(u)$ has been assigned a time such that for each $x \in A(u)$ we have that $t(u') < t(x)$, we can assign a time to $u$ that lies between $\max_{a \in B(u)} t(a)$ and $\min_{x \in A(u)} t(u)$. Hence, we can assign times to the transfer vertices of $G$ recursively as follows:

$$t(u) = \frac{\max\limits_{a \in B(u)} t(a) + \min\limits_{x \in A(u)} t(x)}{2}.$$

It should be clear at this point that if a gene tree vertex has been assigned a time, i.e., if it is a leaf, speciation, or transfer vertex, then its assigned time is greater than any time assigned to its proper ancestors, less than any time assigned to its proper descendants, and $t(p(\gamma(u))) < t(u) < t(\gamma(u))$.

It remains for us to assign times to the duplications of $G$. If $u \in \Delta$, then let

$$C(u) = \{v <_G u : \gamma(v) = \gamma(u), v \notin \Delta\} \cup \{\gamma(u)\}.$$

Clearly, we have already assigned times to all members of $C(u)$ for any $u \in \Delta$. We now recursively assign times to the duplication vertices of $G$ by

$$t(u) = \begin{cases} t(\text{root}(S)) - 1 & \text{if } u = \text{root}(G),\ \gamma(u) = \text{root}(S), \\ \dfrac{t(p(\gamma(u))) + \min\limits_{a \in C(u)} t(a)}{2} & \text{if } u = \text{root}(G),\ \gamma(u) \neq \text{root}(S), \\ \dfrac{\max\{t(p(u)), t(p(\gamma(u)))\} + \min\limits_{a \in C(u)} t(a)}{2} & \text{if } u \neq \text{root}(G). \end{cases}$$

It is now a straightforward verification to check that $2c$ and $2e$ are also satisfied. □

We can check if a DTL-scenario is acyclic in quadratic time. Create a digraph $H$ as follows. Let $V(H) = V(S)$ and $A(H) = E(S) \cup \{\langle p(\gamma(u)), \gamma(v')\rangle : (u, v), (u', v') \in \Xi, v' \leq_G v\}$. The DTL-scenario is acyclic iff $H$ is a DAG. The time complexity is $O(m + n^2)$, where $m = |S|$ and $n = |G|$ (since $|A(H)| = O(m + n^2)$ and checking that a digraph is a DAG can be performed in time proportional to the number of arcs).

It was shown in [35] that finding optimal acyclic DTL-scenarios is NP-hard. The proof can easily be changed to show that finding optimal DTL-scenarios that satisfy the condition used by Jungles is also NP-hard. Hence, the time complexity of Jungles, cannot be polynomial unless $P = NP$. By separating the hard and easy tasks in our methods we obtain an efficient algorithm that works well in practice.

### 4.2 Losses

The process of inferring the number of losses associated with a DTL-scenario was described in [35]. Briefly, for each non-transfer edge $(u, v) \in E(G)$, we count the number of intermediate species tree vertices along the path from $\gamma(u)$ to $\gamma(v)$, i.e., $\gamma(u)$ and $\gamma(v)$ are not counted. If $u$ is a duplication, then we add one to our count if $\gamma(u) \neq \gamma(v)$.

Note that this procedure gives us a lower bound on the number of losses. Also, transfer edges do not contribute to our count of losses in any way; we can only consider losses along transfer edges if we postulate the landing point of the transfer.

## 5 Dynamic Programming Algorithm

In this section, we give an improved dynamic programming algorithm for computing the minimum cost of any DTL-scenario for $S$, $G$, and $\sigma$. In [35], a dynamic programming algorithm with time complexity $O(|S|^2 \cdot |G|)$ was presented for computing the minimum number of duplications and LGTs needed to reconcile $S$ and $G$. The algorithm in this section computes the minimum cost of any DTL-scenario for $S$, $G$, and $\sigma$, where the costs of duplications and LGTs are given by $C_{\mathrm{d}}$ and $C_{\mathrm{t}}$, respectively. The time complexity is improved by an order of magnitude to $O(|S| \cdot |G|)$.

The idea is to use two arrays, **below** and **outside**, both of size $|G| \times |S|$, to keep track of the minimum costs of reconciling subtrees of $G$ with subtrees of $S$. For a gene tree vertex $u$ and species tree vertex $x$, $\mathbf{below}[u, x]$ is the minimum cost of any DTL-scenario for $G_u$ and $S$ where $u$ is mapped by $\gamma$ to some vertex in the subtree $S_x$. For a gene tree vertex $u$ and species tree vertex $x$, $\mathbf{outside}[u, x]$ is the minimum cost of any DTL-scenario for $G_u$ and $S$ given that $u$ is mapped by $\gamma$ to a species tree vertex incomparable to $x$. By computing the entries in the order specified in the algorithm in Figure 6, the time complexity stated above is achieved. The algorithm can be proved correct along the same lines as the proof in [35] and we omit the proof here.

## 6 Parametric Tree Reconciliation

In this section, we develop an algorithm for exploring the space of duplication and LGT costs. Given a gene tree $G$ and a corresponding species tree $S$, there is a finite set of DTL-scenarios that reconcile $S$ and $G$. We associate with each DTL-scenario a pair of natural numbers, $(d, t)$, called the event count pair of the DTL-scenario, where $d = |\Delta|$ is the number of duplications and $t = |\Xi|$ is the number of LGTs of the scenario. We define a partial order on event count pairs as

```
1: below ← Array[1..|V(G)|, 1..|V(S)|] initialized to ∞
2: outside ← Array[1..|V(G)|, 1..|V(S)|] initialized to ∞
3: for all u ∈ V(G) in postorder do
4:     for all x ∈ V(S) in postorder do
5:         if u ∈ L(G) then
6:             if σ(u) ≤_S x then
7:                 below[u, x] ← 0
8:             end if
9:         else
10:            v, w ← children of u
11:            d ← C_d + below[v, x] + below[w, x]
12:            t_v ← C_t + outside[v, x] + below[w, x]
13:            t_w ← C_t + outside[w, x] + below[v, x]
14:            if x ∈ L(S) then
15:                below[u, x] ← min{d, t_v, t_w}
16:            else
17:                y, z ← children of x
18:                s ← min{below[v, y] + below[w, z], below[w, y] + below[v, z]}
19:                below[u, x] ← min{d, s, t_v, t_w, below[u, y], below[u, z]}
20:            end if
21:        end if
22:    end for
23:    for all x ∈ V̊(S) in preorder do
24:        y, z ← children of x
25:        outside[u, y] ← min{outside[u, x], below[u, z]}
26:        outside[u, z] ← min{outside[u, x], below[u, y]}
27:    end for
28: end for
```

Figure 6: The dynamic programming algorithm for computing the minimum cost of reconciling $S$ and $G$.

follows. We say that $(d,t) \leq (d',t')$ when $d \leq d$ and $t \leq t$. Two event count pairs are incomparable iff $d < d'$ and $t > t'$, or $d > d'$ and $t < t'$. An event count pair $(d,t)$ is minimal among a set of event count pairs if there is no event count pair $(d',t')$ such that $(d',t') < (d,t)$. An event count pair is globally minimal if it is minimal in the set of all event count pairs for $S$ and $G$. Note that for any choice of parameters, $C_{\mathrm{d}}$ and $C_{\mathrm{t}}$, only DTL-scenarios with globally minimal event count pairs can be optimal.

Our goal in this section is to develop an efficient algorithm for enumerating all globally minimal event count pairs whose associated DTL-scenarios are optimal for some choice of the parameters $C_{\mathrm{d}}$ and $C_{\mathrm{t}}$, together with a description of the parameter space under which such DTL-scenarios are optimal.

Using the same dynamic programming approach as in section 5, we can compute, in time $O(|G|^3 \cdot |S|)$, the set of globally minimal event count pairs. We use two arrays **below** and **outside**, both of size $|G| \times |S|$, and compute the entries by traversing the gene tree vertices and species tree vertices as specified in the algorithm in Figure 7. Let $u$ and $x$ be a gene tree vertex and a species tree vertex, respectively. If $\beta$ is the set of all event count pairs associated with the DTL-scenarios for $S$ and $G_u$ such that $\gamma(u) \leq_S x$, then the entry $\mathbf{below}[u,x]$ is the set of all minimal event count pairs of $\beta$. Similarly, if $\omega$ is the set of all event count pairs associated with the DTL-scenarios for $S$ and $G_u$ such that $\gamma(u)$ is incomparable to $x$, then $\mathbf{outside}[u,x]$ is the set of all minimal event count pairs of $\omega$. The set of globally minimal event count pairs for $S$ and $G$ is then $\mathbf{below}[\mathrm{root}(G), \mathrm{root}(S)]$.

We now describe how to compute the region of the parameter space in which the DTL-scenarios associated with a certain event cost pair in $\mathbf{below}[\mathrm{root}(G), \mathrm{root}(S)]$ are optimal. As we will see shortly, it is possible that no such region exists for some event cost pairs.

Since the set of optimal DTL-scenarios depends only on the relative sizes of $C_{\mathrm{d}}$ and $C_{\mathrm{t}}$, we will assume that

$$C_{\mathrm{d}} + C_{\mathrm{t}} = 1.$$

Hence, we can view the parameter space as the interval $[0,1]$, say for the duplication cost $C_{\mathrm{d}}$.

**Theorem 2.** *Given $S$, $G$, and $\sigma$, let the globally minimal event count pairs be*

$$(d_1,t_1),(d_2,t_2),\ldots,(d_n,t_n),$$

*where $d_i < d_j$ for $i < j$. A DTL-scenario for $S$, $G$, and $\sigma$ such that $|\Delta| = d_i$ and $|\Xi| = t_i$ is optimal when*

$$\max(\{M_{ij} : j > i\} \cup \{0\}) \leq C_{\mathrm{d}} \leq \min(\{M_{ij} : j < i\} \cup \{1\}),$$

*where* $M_{ij} = \dfrac{t_j - t_i}{d_i - d_j + t_j - t_i}$.

```
1: below ← Array[1..|V(G)|, 1..|V(S)|] initialized to ∅
2: outside ← Array[1..|V(G)|, 1..|V(S)|] initialized to ∅
3: for all u ∈ V(G) in postorder do
4:     for all x ∈ V(S) in postorder do
5:         if u ∈ L(G) then
6:             if σ(u) ≤_S x then
7:                 below[u, x] ← {(0, 0)}
8:             end if
9:         else
10:            v, w ← children of u
11:            below[u, x] ← {d1 + d2 + 1, t1 + t2 : (d1, t1) ∈ below[v, x],
                                                     (d2, t2) ∈ below[w, x]}
                           ∪ {d1 + d2, t1 + t2 + 1 : (d1, t1) ∈ below[v, x],
                                                     (d2, t2) ∈ outside[w, x]}
                           ∪ {d1 + d2, t1 + t2 + 1 : (d1, t1) ∈ below[w, x],
                                                     (d2, t2) ∈ outside[v, x]}
12:            if x ∉ L(S) then
13:                y, z ← children of x.
14:                below[u, x] ← below[u, x]
                               ∪ {(d1 + d2, t1 + t2) : (d1, t1) ∈ below[v, y],
                                                       (d2, t2) ∈ below[w, z]}
                               ∪ {(d1 + d2, t1 + t2) : (d1, t1) ∈ below[v, z],
                                                       (d2, t2) ∈ below[w, y]}
15:            end if
16:        end if
17:        Remove from below[u, x] all non-minimal event counts
18:    end for
19:    for all x ∈ V̊(S) in preorder do
20:        y, z ← children of x
21:        outside[u, y] ← outside[u, x] ∪ below[u, z]
22:        outside[u, z] ← outside[u, x] ∪ below[u, y]
23:        Remove from outside[u, y] and outside[u, z] all non-minimal event
           counts
24:    end for
25: end for
```

Figure 7: The dynamic programming algorithm for computing all globally minimal event count pairs.

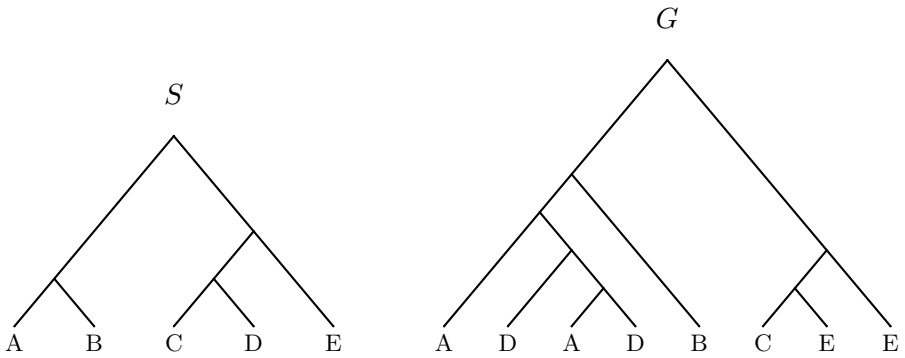



Figure 8: The species tree and the corresponding gene tree used in the example in the main text.

*Proof.* The cost of a DTL-scenarios associated with $(d_i, t_i)$ is less than or equal to the cost of a DTL-scenario associated with $(d_j, t_j)$, $i < j$, when

$$d_i C_\mathrm{d} + t_i C_\mathrm{t} \leq d_j C_\mathrm{d} + t_j C_\mathrm{t},$$

which after substitution of $C_\mathrm{t}$ by $1 - C_\mathrm{d}$ and some rearrangements becomes

$$C_\mathrm{d} \geq \frac{t_j - t_i}{d_i - d_j + t_j - t_i} = M_{ij}.$$

Note that $M_{ij} = M_{ji}$ (just multiply the numerator and denominator by $-1$). We can now conclude, by symmetry, that the cost of DTL-scenarios associated with the event count pair $(d_i, t_i)$ are optimal only when

$$\max(\{M_{ij} : j > i\} \cup \{0\}) \leq C_\mathrm{d} \leq \min(\{M_{ij} : j < i\} \cup \{1\}).$$

□

As an example, take the species tree and gene tree in Figure 8. Using the algorithm of Figure 7, we find that the globally minimal event count pairs are

$$\begin{aligned}(d_1, t_1) &= (0, 4)\\ (d_2, t_2) &= (1, 3)\\ (d_3, t_3) &= (2, 2)\\ (d_4, t_4) &= (4, 1)\\ (d_5, t_5) &= (5, 0).\end{aligned}$$

The $M_{ij}$s are shown below as a matrix:

$$\begin{pmatrix} - & \frac{1}{2} & \frac{1}{2} & \frac{3}{7} & \frac{4}{9} \\ \frac{1}{2} & - & \frac{1}{2} & \frac{2}{5} & \frac{3}{7} \\ \frac{1}{2} & \frac{1}{2} & - & \frac{1}{3} & \frac{2}{5} \\ \frac{3}{7} & \frac{2}{5} & \frac{1}{3} & - & \frac{1}{2} \\ \frac{4}{9} & \frac{3}{7} & \frac{2}{5} & \frac{1}{2} & - \end{pmatrix}$$

So for example, DTL-scenarios with two duplications and two transfers, i.e., DTL-scenarios associated with the event count pair $(d_2, t_2) = (2, 2)$, are optimal when

$$\frac{2}{5} = \max\{0, \frac{1}{3}, \frac{2}{5}\} \leq C_{\mathrm{d}} \leq \min\{\frac{1}{2}, 1\} = \frac{1}{2}.$$

Note that DTL-scenarios with event count pair $(1, 3)$ are optimal exactly when $C_{\mathrm{d}} = \frac{1}{2}$ and those with event count pair $(4, 1)$ are never optimal for any choice of $C_{\mathrm{d}}$.

# 7 Experimental Results

In order to test the applicability of our algorithms, we performed a series of tests on synthetic data. Species trees and gene trees were generated using a probabilistic model of evolution based on the standard birth-death process. We use a pure birth process to generate species trees with divergence times associated with the vertices. A birth-death process is then used to generate gene trees on the generated species trees. A detailed description of the gene evolution model is given in [44]. In short, the model has three rates, $\delta$, $\tau$, and $\mu$, corresponding to the events of duplication, transfer, and loss, respectively. A single gene starts at the root of the species tree and evolves towards the leaves creating a gene tree in the process. The rates determine the distribution of the times between events. When a gene lineage is exposed to a loss event, the gene is removed and its former parent vertex is suppressed. When a gene lineage is exposed to a duplication event, it is replaced by two separate and independent gene lineages that continue evolving along the same species tree edge. When a gene lineage is exposed to a transfer event, it is replaced by two independent lineages, one of which continues to evolve in the same species tree edge, while the other starts to evolve on a different species tree edge chosen uniformly among those that existed at the time of the transfer event. When a gene lineage reaches a species tree vertex, a speciation occurs: the lineage is replaced by two independent lineages that continue evolving along different outgoing edges

of the species tree vertex. This process continues until it reaches the leaves of the species tree at which point the leaf-mapping function, $\sigma$, is defined in the natural way.

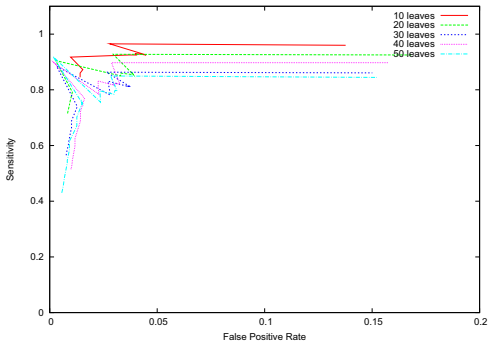
### 7.1 Varying the Costs

First we tested the effect of varying duplication and transfer costs when inferring transfer edges. Having the algorithm for parametric tree reconciliation at our disposal, we are able to get a complete list of all optimal DTL-scenarios using the dynamic programming algorithm in Figure 6.

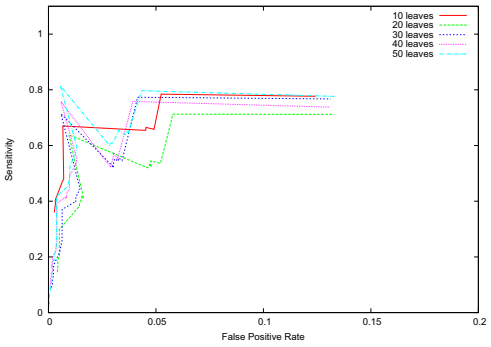
The sizes of the species trees were restricted to between ten and fifty leaves, and the total time from root to leaf in every species tree was set to one. When generating gene trees, the total birth rate was kept equal to the death rate, i.e., $\delta + \tau = \mu$. We varied the total birth rate, and the transfer rate was set to between 0.05 and 0.95 of the total birth rate. When generating gene trees, we kept track of the edges whose history included at least one transfer event; these edges where classified as transfer edges. To infer transfer edges, we obtained the complete set of optimal DTL-scenarios under all cost schemes. For $C_{\mathrm{t}} \in [0, 1]$, we classify a gene tree edge as a transfer edge if it is classified as a transfer edge in at least half of the DTL-scenarios that are optimal when the transfer cost equals $C_{\mathrm{t}}$. In this way, we are able to compute ROC-curves which show the trade-off between high and low transfer costs in terms of sensitivity and false positive rate. Figure 9 shows ROC-curves obtained when the birth rate was set to 1.0. A birth rate of 0.1 generates very few transfer edges so that both sensitivity and specificity (which is equal to one minus the false positive rate) are close to one. Increasing the birth rate leads to more and more difficult cases and when set to 10, our methods give no better results than expected from chance alone. We note here that a birth rate of 1.0 for a species tree with root-to-leaf time of 1.0 is quite high, and a birth rate of 10 appears to be unrealistic for most biological applications.

### 7.2 Completely Accurate DTL-scenarios

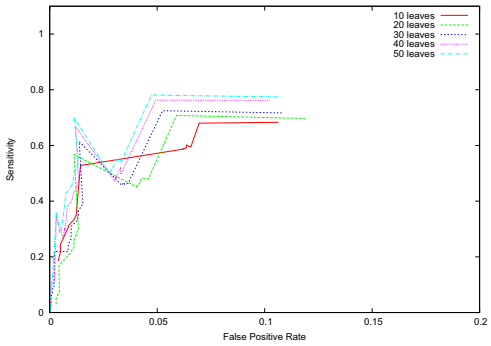
We also tested the ability of our methods to make completely accurate inferences about transfer edges and duplications. More specifically, we wanted to see how often we can expect to find, under some cost scheme, a DTL-scenario that is one hundred percent accurate in classifying both the duplications and the transfer edges. Figure 10 shows how often we find such a DTL-scenario depending on the rate used to generate the gene trees. The size of the species trees varied between 10 and 30 leaves. We should mention that the process of generating gene trees was slightly altered for this test. The general model of gene evolution described in the previous subsection can generate gene trees whose histories cannot be fully



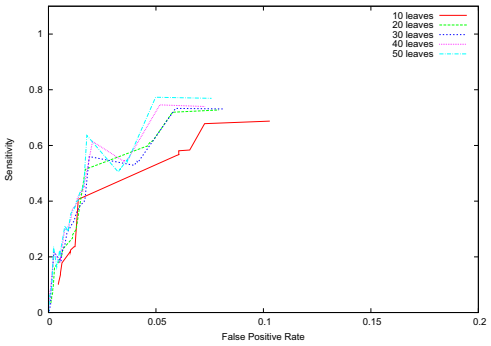

(a) Transfer rate 5% of total birth-rate.



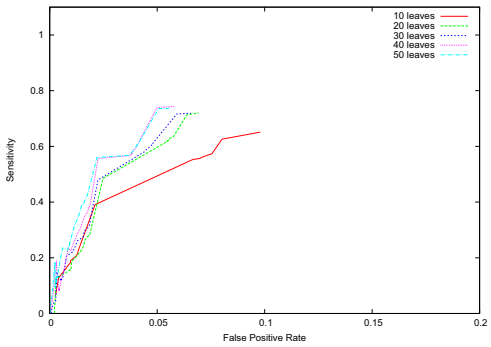

(b) Transfer rate 25% of total birth-rate.




(c) Transfer rate 50% of total birth-rate.




(d) Transfer rate 75% of total birth-rate.




(e) Transfer rate 95% of total birth-rate.

Figure 9: ROC-curves for detection of transfer edges based on 50% majority rule. See the main text for details. Each individual curve is obtained from 200 pairs of species trees and gene trees.

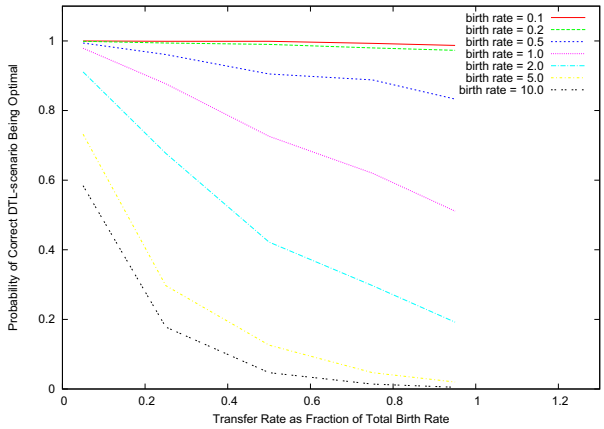



Figure 10: The probability that the DTL-scenario that correctly identifies all transfers and all duplications is an optimal DTL-scenario under some cost scheme is plotted against increasing transfer rate. The root-to-leaf time of the species trees used for this plot was set to 1.0.

described with a DTL-scenario for reasons which were discussed in section 3.2. Therefore, for this test, only gene trees that could adequately be described with a DTL-scenario were kept and the rest were discarded. This simply ensures that there does exist a completely accurate DTL-scenario for each generated pair of species trees and gene tree, and our test is a measure of how often such a DTL-scenario is optimal.

When performing the above test, we also took note of the cost scheme under which the correct DTL-scenario was optimal. To our surprise, we found that the correct DTL-scenario was almost always optimal when $C_{\mathrm{t}} = C_{\mathrm{d}}$, so that we would have obtained basically the same curves as in Figure 10 by only considering that case. In other words, if the true DTL-scenario is optimal under some cost scheme it is almost always optimal when $C_{\mathrm{t}} = C_{\mathrm{d}}$. One explanation for this observation is that the vast majority of optimal DTL-scenarios are optimal exactly when the costs of transfers and duplications are equal.

## References

[1] S. Ohno. *Evolution by gene duplication.* Allen and Unwin, 1970.

[2] M. Lynch and J.S. Conery. The origins of genome complexity. *Science*, 302(5649):1401–1404, Nov 2003.

[3] M. Lynch and J.S. Conery. The evolutionary fate and consequences of duplicate genes. *Science*, 290(5494):1151–1155, Nov 2000.

[4] M. Lynch and J.S. Conery. The evolutionary demography of duplicate genes. *J Struct Funct Genomics*, 3(1-4):35–44, 2003.

[5] M.W. Hahn, T. De Bie, J.E. Stajich, C. Nguyen, and N. Cristianini. Estimating the tempo and mode of gene family evolution from comparative genomic data. *Genome Res*, 15(8):1153–1160, Aug 2005.

[6] J.P. Demuth, T. De Bie, J.E. Stajich, N. Cristianini, and M.W. Hahn. The evolution of mammalian gene families. *PLoS ONE*, 1:e85, 2006.

[7] J.A. Cotton and R.D. Page. Rates and patterns of gene duplication and loss in the human genome. *Proc Biol Sci*, 272(1560):277–283, Feb 2005.

[8] M. Lynch and A. Force. The probability of duplicate gene preservation by subfunctionalization. *Genetics*, 154(1):459–473, Jan 2000.

[9] A. Force, M. Lynch, F.B. Pickett, A. Amores, Y.L. Yan, and J. Postlethwait. Preservation of duplicate genes by complementary, degenerative mutations. *Genetics*, 151(4):1531–1545, Apr 1999.

[10] M. Goodman, J. Czelusniak, G.W. Moore, A.E. Romero-Herrera, and G. Matsuda. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Syst Zool*, 28(2):132–163, 1979.

[11] R. Guigó, I. Muchnik, and T.F. Smith. Reconstruction of ancient molecular phylogeny. *Mol Phylogenet Evol*, 6(2):189–213, Oct 1996.

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

[13] M.J. Sanderson and M.M. McMahon. Inferring angiosperm phylogeny from EST data with widespread gene duplication. *BMC Evol Biol*, 7 Suppl 1:S3, 2007.

[14] RD Page. Genetree: comparing gene and species phylogenies using reconciled trees. *Bioinformatics*, 14(9):819–820, 1998.

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

[16] M.T. Hallett and J. Lagergren. New algorithms for the duplication-loss model. *Proceedings of the fourth annual international conference on Research in Computational Molecular Biology*, pages 138–146, 2000.

[17] J. Lederberg and E. Tatum. Gene recombination in Escherichia coli. *Nature*, page 558, October 1946.

[18] J. Lederberg and E.L. Tatum. Novel genotypes in mixed cultures of biochemical mutants of bacteria. In *Cold Spring Harbor Symp. Quant. Biol*, volume 11, pages 113–114, 1946.

[19] K. Ochiai, T. Yamanaka, K. Kimura, and O. Sawada. Inheritance of drug resistance (and its transfer) between shigella strains and between shigella and e. coli strains. *Nihon Iji Shimpo*, 1861:34, 1959.

[20] H. Ochman, J.G. Lawrence, and E.A. Groisman. Lateral gene transfer and the nature of bacterial innovation. *Nature*, 405(6784):299–304, May 2000.

[21] J.R. Brown. Ancient horizontal gene transfer. *Nat Rev Genet*, 4(2):121–132, Feb 2003.

[22] P.J. Keeling and J.D. Palmer. Horizontal gene transfer in eukaryotic evolution. *Nat Rev Genet*, 9(8):605–618, Aug 2008.

[23] J.P. Gogarten, W.F. Doolittle, and J.G. Lawrence. Prokaryotic evolution in light of gene transfer. *Mol Biol Evol*, 19(12):2226–2238, Dec 2002.

[24] W.F. Doolittle. Phylogenetic classification and the universal tree. *Science*, 284(5423):2124–2129, Jun 1999.

[25] C.R. Woese. On the evolution of cells. *Proc Natl Acad Sci U S A*, 99(13):8742–8747, Jun 2002.

[26] W.F. Doolittle and E. Bapteste. Pattern pluralism and the tree of life hypothesis. *Proc Natl Acad Sci U S A*, 104(7):2043–2049, Feb 2007.

[27] E. Bapteste, E. Susko, J. Leigh, D. MacLeod, R.L. Charlebois, and W.F. Doolittle. Do orthologous gene phylogenies really support tree-thinking? *BMC Evol Biol*, 5(1):33, 2005.

[28] M.T. Hallett and J. Lagergren. Efficient algorithms for lateral gene transfer problems. *Proceedings of the Fifth Annual International Conference on Research in Computational Biology, April 22-25, 2001, Montreal, Canada*, 2001.

[29] L. Nakhleh, D. Ruths, and L.S. Wang. Riata-hgt: A fast and accurate heuristic for reconstructing horizontal gene transfer. *Proceedings of the Eleventh International Computing and Combinatorics Conference (COCOON 05)*, pages 84–93, 2005.

[30] V. Makarenkov and P. Legendre. From a phylogenetic tree to a reticulated network. *J Comput Biol*, 11(1):195–212, 2004.

[31] A. Boc and V. Makarenkov. New efficient algorithm for detection of horizontal gene transfer events. In *Algorithms in Bioinformatics: Third International Workshop, WABI 2003, Budapest, Hungary*, page 190. Springer, 2003.

[32] M.A. Charleston. Jungles: a new solution to the host/parasite phylogeny reconciliation problem. *Math Biosci*, 149(2):191–223, May 1998.

[33] R.D.M. Page and M.A. Charleston. Trees within trees: phylogeny and historical associations. *Trends Ecol Evol*, 13(9):356–359, 1998.

[34] M. Hallett, J. Lagergren, and A. Tofigh. Simultaneous identification of duplications and lateral transfers. *Proceedings of the eighth annual international conference on Research in Computational Molecular Biology*, pages 347–356, 2004.

[35] A. Tofigh, M. Hallett, and J. Lagergren. Simultaneous identification of duplications and lateral transfers. *IEEE ACM Trans Comput Biol Bioinformatics*, 2009.

[36] I. Wapinski, A. Pfeffer, N. Friedman, and A. Regev. Natural history and evolutionary principles of gene duplication in fungi. *Nature*, 449(7158):54–61, Sep 2007.

[37] L. Arvestad, A.C. Berglund, J. Lagergren, and B. Sennblad. Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. *Proceedings of the eighth annual international conference on Research in Computational Molecular Biology*, pages 326–335, 2004.

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

[39] B. Sennblad and J. Lagergren. Probabilistic orthology analysis. *submitted*, 2008.

[40] Ö. Åkerborg, B. Sennblad, L. Arvestad, and J. Lagergren. Simultaneous bayesian gene tree reconstruction and reconciliation analysis. *Proc Natl Acad Sci U S A*, 106(14):5714–5719, Apr 2009.

[41] R.K. Azad and J.G. Lawrence. Detecting laterally transferred genes: use of entropic clustering methods and genome position. *Nucleic Acids Res*, 35(14):4629–4639, 2007.

[42] G. Jin, L. Nakhleh, S. Snir, and T. Tuller. Maximum likelihood of phylogenetic networks. *Bioinformatics*, 22(21):2604–2611, Nov 2006.

[43] J.P. Huelsenbeck, B. Rannala, and B. Larget. A Bayesian framework for the analysis of cospeciation. *Evolution*, 54(2):352–364, Apr 2000.

[44] A. Tofigh, J. Sjöstrand, B. Sennblad, Arvestad L., and Lagergren J. Preliminary: A new model for gene duplication, lgt, loss, rate variation, and sequence evolution. Manuscript, 2009.

[45] David G. Kendall. On the generalized "birth-and-death" process. *Ann Math Statistics*, 19:1–15, 1948.

[46] M. Csűrös and I. Miklós. A probabilistic model for gene content evolution with duplication, loss and horizontal transfer. In *In Tenth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 206–220. Springer, 2006.

[47] W.M. Fitch and T.F. Smith. Optimal sequence alignments. *Proc Natl Acad Sci U S A*, 80(5):1382–1386, Mar 1983.

[48] D. Gusfield, K. Balasubramanian, and D. Naor. Parametric optimization of sequence alignment. *Algorithmica*, 12(4):312–326, 1994.

[49] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.

[50] O. Eulenstein, B. Mirkin, and M. Vingron. Duplication-based measures of difference between gene and species trees. *J Comput Biol*, 5(1):135–148, Spr 1998.

[51] J.P. Doyon, C. Chauve, and S. Hamel. Algorithms for exploring the space of gene tree/species tree reconciliations. *Comparative Genomics: International Workshop, Recomb-CG 2008*, pages 1–13, 2008.

# III

# Detecting LGTs using a novel probabilistic model integrating duplications, LGTs, losses, rate variation, and sequence evolution

A. Tofigh, J. Sjöstrand, B. Sennblad, L. Arvestad, and J. Lagergren

**Abstract**

The debate over the prevalence of lateral gene transfers (LGTs) has been intense. There is now to a large extent consensus around the view that LGT is an important evolutionary force as well as regarding its relative importance across species. This consensus relies, however, mainly on studies of individual gene families.

Up until now, the gold standard for identifying LGTs has been phylogenetic methods where LGTs are inferred from incongruities between a species tree and an associated gene tree. Even in cases where there is evidence of LGT, several concerns have often been raised regarding the significance of the evidence. One common concern has been the possibility that other evolutionary events have caused the incongruities. Another has been the significance of the gene trees involved in the inference; there may for instance be alternative, almost equally likely, gene trees that do not provide evidence for LGT. Independently of these concerns, there has been a need for methods that can be used to quantitatively characterize the level of LGT among sets of species, but also for methods able to pinpoint where in the species tree LGTs have occurred.

Here, we provide the first probabilistic model capturing gene duplication, LGT, gene loss, and point mutations with a relaxed molecular clock. We also provide all fundamental algorithms required to analyze a gene family relative to a given species tree under this model. Our algorithms are based on Markov chain Monte Carlo (MCMC) methodology but build also on techniques from numerical analysis and involve dynamic programming (DP).

## 1 Introduction

The importance and prevalence of lateral gene transfers (LGT) have been debated intensely. The interest in LGT is partly explained by its capacity to transfer pathogenic elements and antibiotic resistance between bacteria, but also the concern that it could transfer, e.g., pesticide resistance from genetically modified crops to other plants.

Transformation, transduction, and conjugation are the mechanisms through which lateral gene transfer can be accomplished among bacteria. Especially transformation has played a pivotal role in several ground-breaking biological experiments. Although DNA had not yet been identified as the carrier of genetic information, pneumococcal strains were

observed to be possible to transform by Griffith in 1928 [16]. Later, the Avery-MacLeod-McCarty experiment showed that DNA is the substance causing bacterial transformation. The possibility of lateral gene transfers among bacteria was realized already in 1946 [27, 28] and demonstrated to occur between different bacterial species in 1959 [31]. A number of studies have established that LGT occurs among prokaryotes, see for example [32] and [4]. Evidence has also been presented for the occurrence of lateral gene transfers from prokaryotes to eukaryotes and even between eukaryotes, see [23] for a recent review.

There has been an intense debate concerning the relative benefits of different methods for phylogenetic tree reconstruction. Today, however, it is common to describe the development of phylogeny algorithms as a progression starting in 1965 with parsimony methods [5, 26, 12], continuing with Maximum Likelihood (ML) methods introduced by Felsenstein [11], and where the most recent contribution is Bayesian methods [21].

The first phylogenetic incongruence methods were constructed to identify gene duplications based on the parsimony principle. Goodman *et al.* [15] pioneered the field by introducing the term *reconciliation* for an embedding of a gene tree into a species tree explaining the evolution of the former. In later contributions, parsimony-based phylogenetic incongruence methods have been described for LGT alone [19], but also for the combination of gene duplications and LGT [18]. The application of Kishino-Hasegawa tests in [30] is another example of a phylogenetic incongruence method for LGT.

The statistical significance of the investigated phylogenetic trees has been a common concern in the context of phylogenetic incongruence methods. Recently, partly prompted by such concerns, Bayesian phylogenetic incongruence methods were developed for duplication analysis. In [37], the GSR model was presented; it is a probabilistic model integrating gene duplication, sequence evolution, and a relaxed molecular clock for substitution rates. Based on the GSR model and using Markov Chain Monte Carlo (MCMC) methodology, a Bayesian analysis tool, PrIME-GSR, was constructed, which takes a known species tree into account and performs simultaneous gene tree reconstruction and reconciliation.

The extreme view that LGT hardly exists (implying that discrepancies between gene and species trees are due to random effects or to insufficiently sophisticated tree reconstruction methods, or possibly due to other events such as duplications) has lost most of its supporters, and instead, LGT is recognized as a major evolutionary force. In fact, due to its prevalence among prokaryotes, the appropriateness of using trees to represent the evolution of some sets of species has been questioned [14, 7, 36], see also [8] and references therein. Here, we will adopt an intermediate view that has emerged in recent years with respect to prokaryotic evolution, namely, that although LGTs are common, they occur with a frequency which is sufficiently low to render tree based representations of organismal evolution meaningful [3].

In the context of hosts and parasites, Huelsenbeck *et al.* developed a Bayesian framework for the detection of host switching using MCMC and taking advantage of sequence information from the host and the parasite species [20]. The model in [20] assumes a one-to-one correspondence between hosts and parasites and does not consider duplications. To our knowledge no probabilistic phylogenetic method has been proposed for simultaneous analysis of duplications and LGTs, although a probabilistic model based on the birth-death process [24] was used in [18] to generate synthetic data, and a similar model was used in [6] to estimate gene family sizes. However, the former never analyzed data

with respect to the model; the latter was only concerned with gene family size, not trees, and LGTs were modeled by introduction events without any explicit points of origin.

We provide the first probabilistic model capturing gene duplication, LGT, gene loss, and point mutations with a relaxed molecular clock. We also provide all fundamental algorithms required to analyze a gene family relative to a given species tree under this model. In the next section, our probabilistic model is presented. In Section 3, we describe an MCMC approach for estimating the posterior distribution of our model. It turns out that computing the generation probability of a gene tree $G$ with edge lengths $l$, $\Pr[G, l|\theta]$ (where $\theta$ consists of parameters of the model), is crucial. We carefully describe how this probability can be expressed, and also, how it can be approximated by introducing discretization points in the species tree. Section 4 contains derivations of differential equations for several important distributions, for instance the probability of extinction. The corresponding distributions can be evaluated at the discretization points using numerical techniques. Also in section 4, we describe how a dynamic programming (DP) algorithm can be constructed for the approximation of $\Pr[G, l|\theta]$ by taking advantage of differential equations, which are also formulated in the section. Differential equations and algorithms that enable approximation of the probability that $G, l$ has been generated using $k$ LGTs are presented in Section 5. Finally, preliminary results from experiments on synthetic data are presented in Section 6

# 2 A new model for duplication, LGT, loss, rate variation, and sequence evolution

The *duplication-transfer-loss gene sequence evolution model with iid rates across gene tree edges*, which we denote DTLSR, is a joint generalization of models used in [18] (which are here described for the first time) and the GSR model [37]. DTLSR integrates the following probabilistic sub-models, which will be described more fully below:

1. A probabilistic duplication-transfer-loss model (DTL-model) describing a gene evolving over a species tree through gene duplication, LGT, and gene loss, thereby generating a gene tree.

2. A substitution rate model describing rate variation over the gene tree.

3. A sequence evolution model describing how nucleotide substitutions occur.

Let the species tree $S$ and the gene tree $G$ generated by the duplication-transfer-loss process be planted trees, i.e., trees with a root of degree one. These trees also have divergence times associated with their vertices. Because $S$ and its divergence times are considered given, they will be omitted from our notation for probabilities, i.e., $\Pr[\cdot|S]$ will be written $\Pr[\cdot]$.

A gene tree vertex represents either a speciation, a duplication, or an LGT event; the divergence time for a speciation vertex is given by the corresponding species tree vertex, while the divergence time for a duplication or an LGT vertex is given by the duplication-transfer-loss process. Divergence times associated with vertices of a tree induce edge times in the natural way.

We use the substitution rate model in order to obtain a relaxed molecular clock [35, 25, 1, 34, 9, 29, 33], which allows for more biological realism. The substitution rate model also turns out to facilitate a more efficient and more accurate MCMC implementation. In the next three subsections, we briefly describe each of the DTLSR sub-models.

## 2.1 Gene duplication, LGT, and gene loss

In the probabilistic DTL-model, a gene tree $G$ evolves over a species tree $S$ with given divergence times. Over any edge $\langle x, y\rangle$ in the species tree, each gene lineage is exposed to gene duplications, LGTs, and gene losses with rates $\delta$, $\tau$, and $\mu$, respectively. That is, in an interval of length $h$ on a species tree edge $\langle x, y\rangle$ the probabilities of a single gene lineage being exposed to a duplication, an LGT, and a loss are, respectively,

$$\delta h, \tau h, \text{ and } \mu h. \tag{1}$$

Moreover, the probability of two or more events happening in such an interval is $o(h)$. When a gene $u$ is exposed to a duplication event, it is replaced by two children, which both continue evolving over the same species tree edge as did $u$. When a gene $u$ is exposed to an LGT, it is replaced by two children: one continuing to evolve over the same species tree edge $\langle x, y\rangle$ as did $u$, and one evolving over another species tree edge chosen uniformly from those concurrent with $\langle x, y\rangle$ at the time of the LGT event. A loss of the gene $u$ removes it from the process as well as from the generated tree, in which also its former parent is suppressed. Each gene lineage reaching a speciation vertex $y$ in $S$ splits into two independent processes, each evolving down distinct outgoing edges of $y$. The process continues recursively down to the leaves where it stops.

The process also generates a *realization* explaining how the gene tree has evolved by mapping each gene tree vertex to a pair with one component being the species tree edge or vertex where the event happened and the other component being the time when the event creating the vertex happened. We will later introduce several types of realizations and the type of realization generated by the process will be called c-realizations. Computing the probability of a given gene tree under the model is non-trivial and we will use a combination of dynamic programming and techniques from numerical analysis to accomplish this task.

## 2.2 Substitution rates

The purpose of the substitution rate model is to transform dated trees with leaves representing extant entities, such trees being necessarily ultra-metric (i.e., all root-to-leaf paths have the same length), into trees consistent with a relaxed molecular clock. This provides a biologically realistic prior distribution for *edge lengths*—the convolution of edge times and substitution rates conventionally used in substitution models. We achieve a relaxed molecular clock by assuming that edge substitution rates are *independently and identically* $\Gamma$*-distributed* variables with mean $m$ and variance $\nu$ [29, 38]. We denote this gamma distribution $\rho$.

Let $l$, $r$, and $t$ denote functions associating an edge length, an edge specific rate, and an edge time, respectively, to each edge of $G$ so that, e.g., $l(u, v)$ is the edge length of the

edge $\langle u, v\rangle$. The relation between lengths, rates, and times over all edges will be denoted by $l = rt$, or conversely $r = l/t$.

### 2.3 Sequence evolution

Each edge in the gene tree has, as explained above, been assigned an *edge length* by the duplication-transfer-loss process and the substitution rate process. Sequence evolution over the gene tree with these edge lengths can be modeled using any of the standard substitution models used in phylogenetics [11].

## 3 MCMC and discretizing the gene tree probability

MCMC is commonly used to estimate the posterior of phylogenetic trees for given gene sequences [21]. In this application of the MCMC methodology, it is natural to let the states consist of trees with edge lengths and additional parameters. When considering to use MCMC to estimate posterior probabilities under the GSR model [37], the most immediate idea is to also include a reconciliation of the gene and species tree (which explains how the gene tree evolved by mapping it into the species tree, the explanation may contain duplications and losses but not LGTs) as a component of the state; this approach would, however, lead to several technical complications. Fortunately, it is possible to evade these problems by estimating an integral over all reconciliations [37]. Here we will use a similar approach, although in our case, estimating the integral is significantly harder due to the inclusion of LGTs.

To simplify notation, let $\theta = (\delta, \tau, \mu, m, \nu)$ denote the parameters of the DTLSR model (there may also be additional parameters associated with the sequence evolution model, but we omit these from the present notation). Our Markov chain will have states of the form $(G, l, \theta)$ where $G$ is a gene tree, $l$ denotes edge lengths, and $\theta$ denotes parameters of the DTLSR model. Ratios between posterior probabilities of the form $p[G, l, \theta|D]$ need to be computed in order to determine acceptance probabilities of proposed states in our Markov chain. This posterior probability can be rewritten as follows:

$$p[G, l, \theta|D] = \frac{\Pr[D|G, l] \Pr[G, l|\theta]\, p[\theta]}{\Pr[D]}, \tag{2}$$

where the parameters $\theta$ are assigned independent priors (which will be uniform or some other distribution that we can compute). As usual in MCMC estimation of posterior probabilities, the denominators will cancel in any ratio between two such probabilities. Moreover, the factor $\Pr[D|G, l]$ can be computed using the standard DP algorithm introduced by Felsenstein [11]. The last component of our MCMC algorithm for estimating the posterior of the DTLSR model is a procedure to estimate $\Pr[G, l|\theta]$.

In the next subsections, we will show how to estimate $\Pr[G, l|\theta]$ by summing over realizations that only associate gene tree vertices to points from a set of discretization points on the species tree. We will clearly describe the two approximations we make. The expression below is formally incorrect (since our density function is discontinuous at the vertices of $S$) but fits intuitively with the formal description we will give, and it also leads

to a functional MCMC algorithm for estimating posteriors of gene trees. The integration is over realizations $t$ and the summation over discretized realizations:

$$\begin{aligned}\Pr[G, l|\theta] &\approx \int_t p[G, l, t|\theta]dt \\ &= \int_t p[(r = l/t)|m, \nu]\ p[G, t|\delta, \tau, \mu]dt \\ &\approx \sum_t p[(r = l/t)|m, \nu]\ p[G, t|\delta, \tau, \mu].\end{aligned}$$

## 3.1 Definitions

We will now introduce several concepts that will be useful in the rest of the article. When the notation introduced in this subsection is used, the tree will be clear from the context. For each species tree $T$ and each vertex $x \in V(T)$, there will be an associated divergence time $t(x)$. Associated with each edge $\langle x, y\rangle$ of a species tree is the interval $I(x, y) = [t(y), t(x)]$. The leaves of a species tree have divergence time 0 and each internal vertex has divergence time $> 0$. We will assume that all speciations have taken place at distinct times although all our results can easily be modified to allow concurrent speciations.

The following definitions are standard. An edge $\langle x, y\rangle$ has *tail* $x$ and *head* $y$ and it is an *outgoing* edge of $x$. For a pair of edges $e$ and $f$ of the same tree, if the head of $e$ is the tail of $f$, then $e$ is the *parent* of $f$ and $f$ a *child* of $e$. If there is an edge $\langle x, y\rangle$ in the tree $T$, then $x$ is the parent of $y$ and denoted $p_T(y)$. The *proper ancestor* relation is the transitive closure of the parent relation. That $a$ is a proper ancestor of $b$ in the tree $T$ is denoted $a >_T b$, and $b$ is also said to be a proper descendant of $a$. If $a$ equals $b$ or is a proper ancestor of $b$ in $T$, then $a$ is an *ancestor* of $b$ in $T$, which is denoted $a \geq_T b$. Two vertices are said to be comparable if one is a descendant of the other, and incomparable otherwise. Finally, the planted subtree of $T$ containing $u$, its parent $p_T(u)$, and all descendants of $u$ is denoted $T^u$.

## 3.2 A discrete approximation of the probability of a gene tree

In this subsection, we show how to properly express $\Pr[G, l|\theta]$. A key step is to discretize the species tree, which will also give us subintervals of the edges of $S$ in which the discretization points can be considered to be midpoints. We will use two approximation steps in order to compute $\Pr[G, l|\theta]$. The first approximation is an assumption that only one of any two comparable vertices in $G$ can be created by the events occurring in a particular subinterval. The second approximation is obtained by approximating the density function in any point of the subinterval by the density function's value in the subinterval's midpoint.

We will now in two steps introduce discretization points in $S$ to obtain a second species tree $S'$ and then also a third species tree $S''$. Let $S'$ be the tree obtained from $S$ by recursively for each $t \in \{t(x) : x \in V(S)\}$ subdividing each edge $\langle x, z\rangle$ of $S$ such that $t(z) < t < t(x)$ by introducing a new vertex $y$ and letting the divergence time of $y$ be defined by $t(y) = t$. See Figure 1a and 1b for an example.

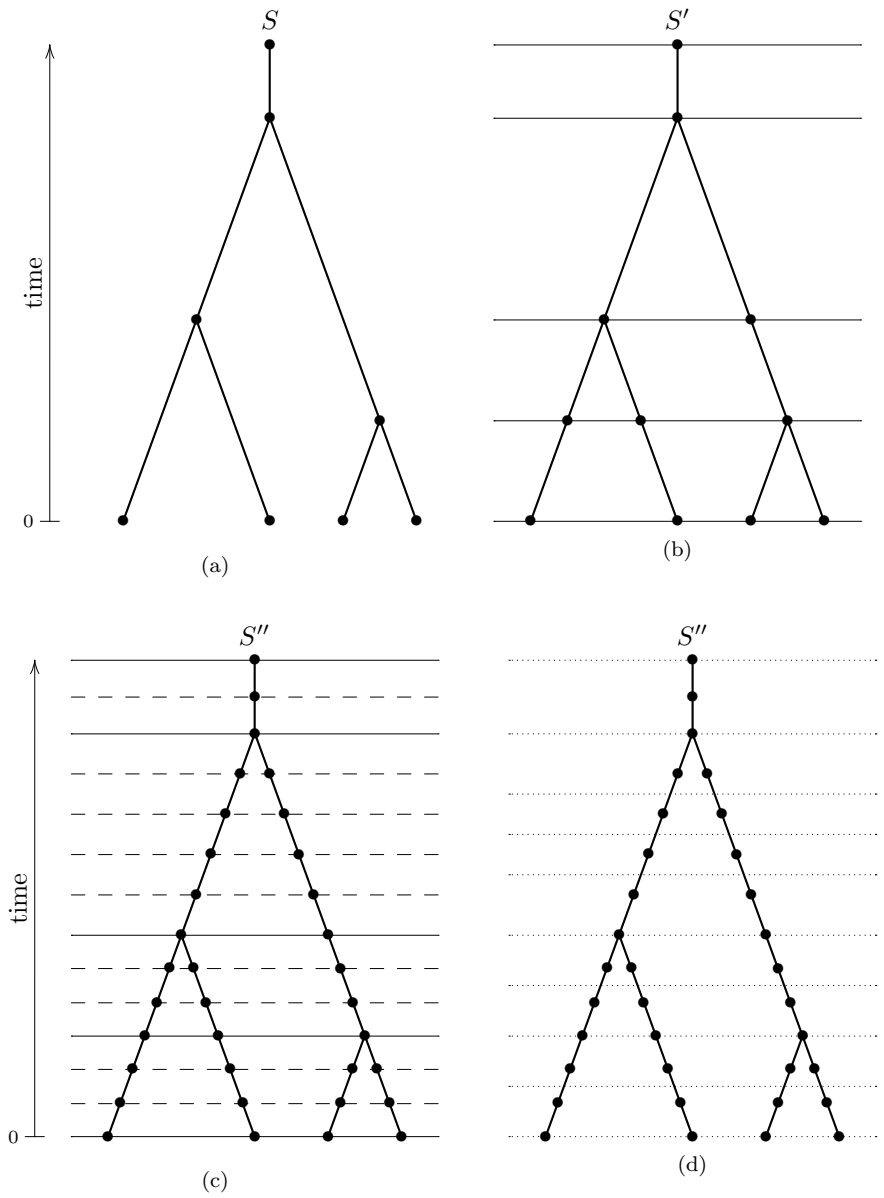



Figure 1: **The subdivisions and subintervals of the species tree.** (a) A species tree $S$. (b) The tree $S'$. (c) The tree $S''$. (d) The subintervals $\Delta$ associated with the vertices of $S''$. Note how each discretization point in $\mathcal{D}$ is a "midpoint" of a subinterval.

We subdivide $S'$ by introducing new vertices at a number of discretization points $\mathcal{D}$. The set of discretization points can in principle be arbitrary, although the accuracy of the algorithm will depend on it. It is, for instance, natural to use all multiples of an interval length $d$ as the set of discretization points, i.e.,

$$\mathcal{D} = \{dk : k \in N^+ \text{ and } dk \leq t(\text{root}(S'))\}.$$

We will for convenience assume that $\mathcal{D}$ and $\{t(x) : x \in V(S')\}$ are disjoint. Let $S''$ be the tree obtained from $S'$ by recursively for each $t \in \mathcal{D}$ subdividing each edge $\langle x, z\rangle$ such that $t(z) < t < t(x)$ by introducing a new vertex $y$ and letting the divergence time of $y$ be defined by $t(y) = t$. See Figure 1c for an example.

A *continuous realization* (c-realization) of $G$ is a function $c : V(G) \to \{\langle e, t\rangle : e \in E(S') \text{ and } t \in I(e)\} \cup \{\langle x, t(x)\rangle : x \in V(S)\}$ such that for each $u >_G v$,

$$c_t(u) > c_t(v),$$

where $c_t(u)$ denotes the projection of the second component of $c(u)$. The projection of the first component of $c(u)$ is denoted $c_V(u)$. A *speciation realization* (s-realization) of $G$ is a function $s : U \to V(S)$ where $U \subseteq V(G)$ such that for each $u, v \in U$, $u >_G v$ implies

$$t(s(u)) > t(s(v)).$$

A *discrete realization* (d-realization) of $G$ is a function $d : V(G) \to (V(S'') \setminus V(S')) \cup V(S)$ such that for each $u >_G v$,

$$t(d(u)) > t(d(v)).$$

Notice that for each edge $e$ of $S''$, there is a unique edge $\langle x, y\rangle$ of $S'$ such that the path in $S''$ between the vertices $x$ and $y$ contains $e$; we say that the edge $\langle x, y\rangle$ *captures* the edge $e$. Analogously for each vertex $z \in V(S'') \setminus V(S')$, there is a unique edge $\langle x, y\rangle$ of $S'$ such that the path in $S''$ between the vertices $x$ and $y$ contains $z$; we say that the edge $\langle x, y\rangle$ *captures* the vertex $z$.

For each vertex $x \in V(S'') \setminus V(S')$ that is captured by the edge $e \in E(S')$, we associate what can be called a subinterval of $e$ as follows. Assume that $y$ is the single child of $x$. First, if $p_{S''}(x) \in V(S')$, define $t_p(x)$ to be $t(p_{S''}(x))$, and otherwise define $t_p(x)$ to be $(t(p_{S''}(x)) + t(x))/2$. Second, if $y \in V(S')$, define $t_c(x)$ to be $t(y)$, and otherwise define $t_c(x)$ to be $(t(x) + t(y))/2$. Finally, let $\Delta(x) = [t_c(x), t_p(x))$ and let $|\Delta(x)|$ denote the length of the the interval $\Delta(x)$, i.e., $|\Delta(x)| = t_p(x) - t_c(x)$. See Figure 1d for an example.

Let $s$ be an s-realization of $G$. A c-realization $c$ of $G$ is a c-extension of $s$ if $c_V|_{c_V^{-1}(V(S))} = s$. Similarly, a d-realization $d$ of $s$ is a d-extension of $s$ if $d|_{d^{-1}(V(S))} = s$. A c-realization is *sparse* if for each $x \in V(S'')$ and each pair of vertices $u, v \in V(G)$,

$$u >_G v \text{ and } c_t(u) \in \Delta(x) \text{ implies } c_t(v) \notin \Delta(x).$$

Let $\mathbf{s_G}$ be the set of s-realizations of $G$. Let $\mathbf{d_G}$ be the set of d-realizations of $G$. For any s-realization of $G$, let $\chi_{\mathbf{c}}(s)$, $\chi_{\mathbf{s}}(s)$, and $\chi_{\mathbf{d}}(s)$ be the sets of c-extensions, sparse c-extensions, and d-extensions of $s$, respectively. Since the vertices of $S$ create discontinuities in the density $p[G, l, c|\theta]$, we express the probability $\Pr[G, l|\theta]$ as the following sum:

$$\sum_{s \in \mathbf{s_G}} \Pr[G, l, s|\theta] = \sum_{s \in \mathbf{s_G}} \int_{c \in \chi_{\mathbf{c}}(s)} p[G, l, c|\theta] dc.$$

We approximate the RHS by

$$\sum_{s \in \mathbf{s_G}} \int_{c \in \chi_{\mathbf{s}}(s)} p[G, l, c|\theta] dc.$$

Notice that, for any c-realization or sparse c-realization $c$,

$$p[G, l, c|\theta] = \prod_{\langle u,v \rangle \in E(G)} p[l(u, v), c(v)|c(u), \theta].$$

As a notational convenience, we define $|\Delta(x)| = 1$ for $x \in V(S')$. Our approximation of the probability $\Pr[G, l|\theta]$ can now be summarized as

$$\begin{aligned}
\Pr[G, l|\theta] &= \sum_{s \in \mathbf{s_G}} \int_{c \in \chi_{\mathbf{c}}(s)} p[G, l, c|\theta] dc \\
&\approx \sum_{s \in \mathbf{s_G}} \int_{c \in \chi_{\mathbf{s}}(s)} p[G, l, c|\theta] dc \\
&= \sum_{s \in \mathbf{s_G}} \int_{c \in \chi_{\mathbf{s}}(\mathbf{s})} \prod_{\langle u,v \rangle \in E(G)} p[l(u, v), c(v)|c(u), \theta] dc \\
&\approx \sum_{s \in \mathbf{s_G}} \sum_{d \in \chi_{\mathbf{d}}(\mathbf{s})} \prod_{\langle u,v \rangle \in E(G)} p[l(u, v), d(v)|d(u), \theta] \cdot |\Delta(d(v))|. \qquad (3)
\end{aligned}$$

In the next section, we will show how to compute the right hand side of the above equation.

# 4 Computing the probability of a gene tree using DP

In this section, we will show how to compute the RHS of (3) using DP. In the DP algorithm, two distributions will turn out to be useful. First, the probability of extinction, for which we will derive differential equations in Subsection 4.1. Second, in Subsection 4.2, we will derive differential equations for the probability of a single gene $u$ evolving, between two points in the species tree, to a single descendant $v$ that may give rise to descendants in the extant species ($u$ may also have other descendants contemporary to $v$ but these will go extinct before reaching the leaves of the species tree). In the last subsection, we derive a DP algorithm for computing (3) from differential equations.

Let $\phi = \delta + \tau + \mu$. When the notation introduced in this paragraph is used it will be clear from the context whether the species tree concerned is $S'$ or $S''$. Two vertices $x$ and $y$ are said to be *contemporary* if $t(x) = t(y)$. Two edges $\langle x, y \rangle$ and $\langle x', y' \rangle$ are said to be *contemporary* if $t(x) = t(x')$ and $t(y) = t(y')$ (in fact, in both $S'$ and $S''$, $t(x) = t(x')$ implies $t(y) = t(y')$ and the other way around). An *edge generation* is a maximal set of pairwise contemporary edges. The edge generation containing $e$ is denoted $\mathcal{G}_E(e)$ and the edges contemporary to $e$, i.e., $\mathcal{G}_E(e) \setminus \{e\}$, is denoted $\mathcal{C}_E(e)$. For two edges $e, f$, if $e$ is the parent of $f$, then $\mathcal{G}_E(e)$ is the *parental generation* of $\mathcal{G}_E(f)$.

## 4.1 The probability of extinction

In this subsection, we derive differential equations for the probability of extinction, which can be solved numerically. For $e \in E(S')$ and $t \in I(e)$, let $Q_e(t)$ be the probability of extinction when starting with a single gene at time $t$ on edge $e$. The following system of differential equations follow from standard techniques for Poisson processes [10, 2] and the fact that when an LGT occurs, the edge to which the transfer is made is chosen uniformly:

$$\frac{\mathrm{d}}{\mathrm{d}t}Q_e(t) = \delta(Q_e(t))^2 + \tau\left(\sum_{f\in\mathcal{C}_E(e)}\frac{1}{|\mathcal{C}_E(e)|}Q_e(t)Q_f(t)\right) + \mu - \phi Q_e(t). \tag{4}$$

For $e = \langle x, y\rangle \in E(S')$, the initial values for the system of equations above are given by

$$Q_e(t(y)) = \begin{cases} 0 & \text{if } y \text{ is a leaf,} \\ Q_f(t(y)) & \text{if } f \text{ is the single child of } e, \\ Q_f(t(y))Q_g(t(y)) & \text{if } f \text{ and } g \text{ are the two children of } e. \end{cases}$$

For one generation of edges of $S'$, the systems of equations for $Q_e$ can be solved using standard Runge-Kutta numerical solvers [17] once the systems for proper descendant generations have been solved. That is, we can solve these equations first for the generation of edges incident to the leaves and then continue upwards to the root of the species tree. For the edge generation $\mathcal{G}_E(e)$, we solve $Q_e(t)$ for all $t \in \{t(x) : x \in V(S'')\} \cap I(e)$.

## 4.2 The probability of exactly one mortal descendant

In this subsection, we apply the same approach as in the previous subsection. In this case, we are interested in the probability of a single gene $u$ evolving, between to points in the species tree, to a single descendant $v$ that may give rise to descendants in the extant species ($u$ may also have other descendants contemporary to $v$ but none of these should give rise to extant descendants).

By a *ghost* we mean a gene in the probabilistic DTL-model that will not have any descendants among the leaves of the species tree. In contrast, a *mortal* is a gene that may or may not yield descendants among the leaves of the species tree. For a pair of edges $e, f$ of $S'$ such that $s \in I(e)$, $t \in I(f)$, and $t < s$, define $Q_{ef}(s,t)$ as the probability of starting on $e$ at time $s$ and having one mortal in $f$ at time $t$ and all other descendants at time $t$ being ghosts.

Let $e, f \in E(S')$ be two contemporary edges. As before, the following system of differential equations can be obtained using standard techniques:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}s}Q_{ef}(s,t) &= 2\delta Q_e(s)Q_{ef}(s,t) - \phi Q_{ef}(s,t) \\ &\quad + \tau\sum_{g\in\mathcal{C}_E(e)}\frac{1}{|\mathcal{C}_E(e)|}\Big(Q_{gf}(s,t)Q_e(s) + Q_{ef}(s,t)Q_g(s)\Big).\end{aligned}$$

The initial values for the above equations are given by

$$Q_{ef}(t,t) = \begin{cases} 1 & \text{if } e = f, \\ 0 & \text{otherwise.} \end{cases}$$

For one generation of edges of $S'$, the systems of equations for $Q_{ef}$ and $Q_e$ can be solved together using standard Runge-Kutta numerical solvers. For the edge generation $\mathcal{G}_E(e)$, we solve $Q_{fg}(s,t)$ for all $f,g \in \mathcal{G}_E(e)$ and $s,t \in \{t(x) : x \in V(S'')\} \cap I(e)$.

We will now show how to compute $Q_{ef}$ when $e$ is a proper ancestor of $f$, i.e., when $e$ and $f$ belong to different edge generations of $S'$. Let $e = \langle x, y\rangle$ and assume that $g$ is the unique edge that is contemporary with $e$ and has two children $g'$ and $g''$. For any $s \in I(e)$ and $t \in I(f)$, $Q_{ef}(s,t)$ can be written

$$Q_{ef}(s,t) = Q_{eg}(s,t(y))\Big(Q_{g'f}(t(y),t)Q_{g''}(t(y)) + Q_{g''f}(t(y),t)Q_{g'}(t(y))\Big) + \sum_{h \in \mathcal{C}_E(g)} Q_{eh}(s,t(y))Q_{hf}(t(y),t).$$

The above equations are solved for all $s \in \{t(x) : x \in V(S'')\} \cap I(e)$ and all $t \in \{t(x) : x \in V(S'')\} \cap I(f)$. These equations can be solved for all pairs of edge generations and discretization points recursively from the leaves of the species tree towards the root.

## 4.3 The final recursion

In this subsection, we derive a DP algorithm for computing (3) from the differential equations in the previous subsections.

We will need to compute the probability of extinction at the vertices of $S''$ and the probability of evolving to exactly one mortal between any pair of vertices in $S''$. For $e = \langle y, z\rangle \in E(S'')$ and $x \in V(S'')$ which is the head of the edge $f$ in $S''$ and satisfies $t(x) \leq t(z)$, we define $p_{11}(e,x)$ as follows

$$p_{11}(e,x) = Q_{e'f'}(t(y),t(x)),$$

where $e'$ and $f'$ are the edges of $S'$ that capture $e$ and $f$, respectively.

Now, to compute the probability of a gene tree $G$, we sum the probabilities of every possible mapping of the gene tree vertices on the vertices of $S''$. For $x \in V(S'') \setminus L(S'')$ and $u \in V(G) \setminus L(G)$, define $a(x,u)$ as the probability of $G_u$ given that the event creating $u$ occurred at $x$. For $e \in E(S'')$ and $u \in V(G)$, define $s(e,u)$ as the probability of the planted tree $G^u$ when starting at the tail of $e$. These two probabilities can be computed as follows. If $x$ is a speciation, then

$$a(x,u) = s(e,v)s(f,w) + s(e,w)s(f,v),$$

where $e, f$ are the outgoing edges of $x$ and $v, w$ are the children of $u$. If $x$ is not contemporary to any speciations, then

$$a(x,u) = 2\delta s(e,v)s(e,w) + \tau \sum_{f \in \mathcal{C}_E(e)} \frac{1}{|\mathcal{C}_E(e)|}\Big(s(e,v)s(f,w) + s(e,w)s(f,v)\Big),$$

where $e$ is the outgoing edge of $x$. We define $a(x,u)$ to be zero in the two other possible cases, i.e., when $x$ is a leaf or has out-degree one and is also contemporary to a speciation.

For $e = \langle x, y\rangle \in E(S'')$ and $u \in V(G)$, we can compute $s(e,u)$ as follows:

$$s(e,u) = \begin{cases} p_{11}(e,\sigma(u))\rho\left(\frac{l(p(u),u)}{t(x)}\right) & \text{if } u \in L(G), \\ \sum_{z\in\mathcal{Q}(x)} p_{11}(e,z)\rho\left(\frac{l(p(u),u)}{t(x)-t(z)}\right) a(z,u) & \text{otherwise,} \end{cases}$$

where $\mathcal{Q}(x)$ is the set of all vertices $z$ of $S''$ such that $t(z) < t(x)$.

# 5 Counting Transfers

In this section we present differential equations and algorithms that enable approximation of the probability that $G, l$ has been generated using $k$ LGTs. We wish to compute the probability that $G$ is generated and that exactly $k$ LGTs has occurred on the paths to the leaves of $G$ during the generation of $G$. In order to accomplish this, we will compute almost the same probabilities as in the previous section, but with the addition of an index $k$ to keep track of the number of LGTs used. In this sense, this section is an extension of the previous section. The probability of extinction is the same as before, since we are not counting the number of LGTs only creating ghosts.

Let $e$ and $f$ be two contemporary edges of $S''$. For $t \le s \in I(e)$, define $Q_{efk}(s,t)$ to be the probability of starting on $e$ at time $s$ having some number of ghosts at time $t$ and, except for these ghosts, having only produced one mortal on $f$ at time $t$ using exactly $k$ LGTs (so there is a path containing $k$ LGTs that end on $f$ at time $t$). For $k > 0$, the following holds

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}Q_{efk}(t) = {} & 2\delta Q_e(s)Q_{efk}(s,t) - \phi Q_{efk}(s,t) \\ & + \tau \sum_{g\in\mathcal{C}_E(e)} \frac{1}{|\mathcal{C}_E(e)|}\Big(Q_{gf(k-1)}(s,t)Q_e(s) + Q_{efk}(s,t)Q_g(s)\Big).\end{aligned}$$

For $k = 0$ a similar expression can be obtained

$$\frac{\mathrm{d}}{\mathrm{d}t}Q_{ef0}(t) \quad = \quad 2\delta Q_e(s)Q_{ef0}(s,t) - \phi Q_{ef0}(s,t) + \tau \sum_{g\in\mathcal{C}_E(e)} \frac{1}{|\mathcal{C}_E(e)|}Q_{ef0}(s,t)Q_g(s).$$

As before, the initial values for the above equations are given by

$$Q_{efk}(t,t) = \begin{cases} 1 & \text{if } e = f \text{ and } k = 0, \\ 0 & \text{otherwise.} \end{cases}$$

We will now show how to compute $Q_{efk}$ when $e = \langle x, y\rangle$ and $f$ belong to different edge generations. Assume that $g$ is the unique edge in $\mathcal{G}_E(e)$ that has two children $g'$ and $g''$. For any edge $h \in \mathcal{C}_E(g)$, let $h'$ denote the unique child of $h$. For any $s \in I(e)$ and

$t \in I(f)$, $Q_{efk}(s,t)$ can be written

$$Q_{efk}(s,t) = \sum_{k'+k''=k} \Bigg( Q_{egk'}(s,t(y)) \Big( Q_{g'fk''}(t(y),t) Q_{g''}(t(y)) + Q_{g''fk''}(t(y),t) Q_{g'}(t(y)) \Big) + \sum_{h \in \mathcal{C}_E(g)} Q_{ehk'}(s,t(y)) Q_{h'fk''}(t(y),t) \Bigg).$$

For $e = \langle y, z \rangle \in E(S'')$ and $x \in V(S'')$ such that $t(x) \leq t(z)$, we define $p_{11k}(e,x)$ as follows

$$p_{11k}(e,x) = Q_{e'f'k}(t(y),t(x)),$$

where $e'$ and $f'$ are the edges of $S'$ that captures $e$ and $f$, respectively.

To compute the probability of a gene tree $G$ with $k$ LGTs, we sum the probabilities of every possible mapping of the gene tree vertices on the vertices of the subdivision $S''$. For $x \in V(S) \setminus L(S)$ and $u \in V(G) \setminus L(G)$, define $a_k(x,u)$ as the probability of generating $G_u$ using exactly $k$ LGTs given that the event creating $u$ occurred at $x$. For $e \in E(S')$ and $u \in V(G)$, define $s_k(e,u)$ as the probability of generating the planted tree $G^u$ using $k$ LGTs when starting at the tail of $e$. These two probabilities can be computed as follows.

If $x$ is a speciation, then

$$a_k(x,u) = \sum_{k'+k''=k} s_{k'}(e,v) s_{k''}(f,w) + s_{k'}(e,w) s_{k''}(f,v),$$

where $e, f$ are the outgoing edges of $x$ and $v, w$ are the children of $u$. If $x$ is not contemporary to any speciations, i.e., contemporary to any vertices of $S$, then

$$a_k(x,u) = \sum_{k'+k''=k} 2\delta s_{k'}(e,v) s_{k''}(e,w) + \tau \sum_{k'+k''=k-1} \sum_{f \in \mathcal{C}_E(e)} \frac{1}{|\mathcal{C}_E(e)|} \Big( s_{k'}(e,v) s_{k''}(f,w) + s_{k''}(e,w) s_{k'}(f,v) \Big),$$

where $e$ is the outgoing edge of $x$. We define $a_k(x,u)$ to be zero in the two other cases, i.e., when $x$ is a leaf or has out-degree one and is also contemporary to a speciation.

For $e = \langle x, y \rangle \in E(S')$ and $u \in V(G)$, we can compute $s_k(e,u)$ as follows:

$$s_k(e,u) = \begin{cases} p_{11k}(e,\sigma(u)) \rho \left( \frac{l(p(u),u)}{t(x)} \right) & \text{if } u \in L(G), \\ \sum_{k'+k''=k} \sum_{z \in \mathcal{Q}(x)} p_{11k'}(e,z) \rho \left( \frac{l(p(u),u)}{t(x)-t(z)} \right) a_{k''}(z,u) & \text{otherwise,} \end{cases}$$

where $\mathcal{Q}(x)$ is the set of all vertices $z$ of $S'$ such that $t(z) < t(x)$.

# 6 Experimental results

In this section, we present results of preliminary experiments performed on synthetic data. For our species tree, we selected a subset of the taxa in the yeast tree which together with

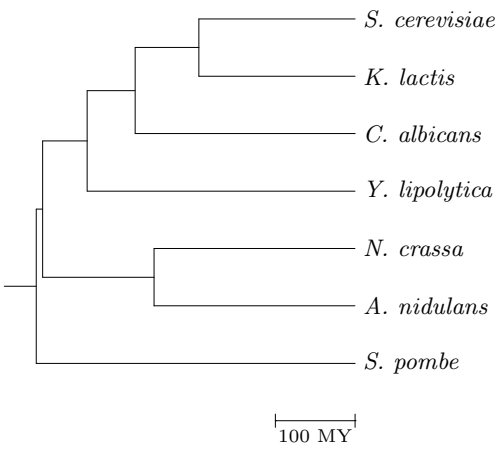



Figure 2: **The yeast tree used in the experiments.** The root-to-leaf time of the tree is approximately 400 million years. In the experiments, the time was rescaled so that the root-to-leaf time became 1.0 and the total birth-rate became 0.17 (relative to the new time scale).

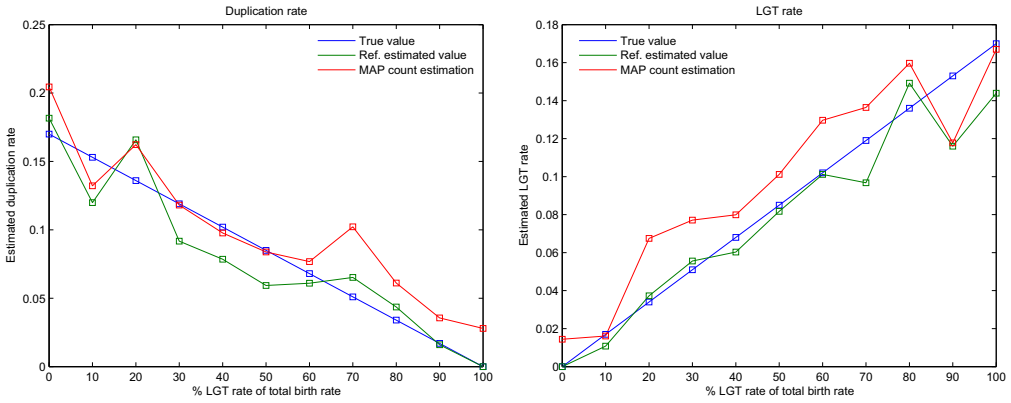
divergence times was presented in [37]. The species tree is shown in Figure 2. The species tree was rescaled so that the root-to-leaf time equaled 1.0. In addition, an edge of length 0.1 preceding the root was introduced in order to allow duplication events to occur prior to the first speciation. Naturally, LGT events may not take place along this edge.

We generated 11 distinct sets of gene trees according to the probabilistic DTL-model, each comprising 100 trees. Analysis of the data from [37] yields an estimated death rate of approximately 0.17. When generating the trees, both the death rate $\mu$ and the total birth rate, i.e., $\delta + \tau$, were kept fixed at 0.17, while the LGT rate $\tau$ varied between 0% and 100% of the total birth rate in steps of 10% increments. All gene trees were produced starting with a single lineage at the earliest point of the species tree.

We used the resulting gene tree topologies and divergence times to generate sequences using the JTT amino acid substitution model [22]. Edge rates were drawn *iid* from a gamma distribution with mean 0.5 and variance 0.1, with no rate variation among sites. The output of this procedure was aligned amino acid sequences of length 1,000.

In order to verify the soundness of our probabilistic model, we analyzed the posterior distributions of the duplication and LGT rates. The gene tree topologies were kept fixed to the true topologies, while the remaining parameters were inferred by the MCMC process.

As we are not yet able to perform automatized convergence testing, we conducted a series of pilot tests to analyze mixing and simulation length requirements. We selected several of the smallest and largest trees from each set and ran three seperate chains with 1,000,000 iterations per tree, sampling every 100th iteration. A small number of trees proved too small for stable performance, each such tree had only two leaves. These were consequently removed from further consideration. Parameter trace plots on remaining trees indicated good mixing. Convergence was evaluated using the Gelman-Rubin test [13],




(a) (b)

Figure 3: **Results of tests performed on synthetic data.** The estimated duplication and LGT rates are plotted against the true rates. The blue curve plots the true rates showing the ideal estimation curves. The green curve shows our reference estimate of the rates using previous knowledge about the true duplication vertices and transfer edges. The red curve is obtained by first estimating the number of events using the posterior distribution of the rates, and analogous to the green curve, using these values to obtain estimates of the rates.

where all parameters had a joint test statistic $\leq 1.02$, with the exception of the LGT rate in one instance reaching 1.08. All tests were conducted with the first 10% of the samples removed as burn-in. We concluded that these settings are sufficient to provide convergence in most cases, and used them in subsequent analyses.

The output of each tree, i.e., the merged chain triplet, was used to estimate the posterior distribution. We then analyzed the marginal distributions of the duplication and LGT rates, and after applying a moderate smoothing kernel, the MAP rate for each gene tree was used to estimate the number of duplication and LGT events.

The true number of duplication vertices and transfer edges divided by the total length of the species tree was used to estimate the birth rates of each gene tree. The average value for each set was used as a reference estimate of the birth rates. Similarly, the estimated number of duplications and LGTs derived from the posterior distributions were divided by the total length of the species tree to obtain estimates of the birth rates from our MCMC procedure. Figure 3 shows the results.

# References

[1] S. Aris-Brosou and Z. Yang. Effects of models of rate evolution on estimation of divergence dates with special reference to the metazoan 18s ribosomal RNA phylogeny. *Syst Biol*, 51(5):703–714, Oct 2002.

[2] N.T.J. Bailey. *The elements of stochastic processes with applications to the natural sciences*. Wiley-Interscience, 1990.

[3] E. Bapteste, E. Susko, J. Leigh, D. MacLeod, R.L. Charlebois, and W.F. Doolittle. Do orthologous gene phylogenies really support tree-thinking? *BMC Evol Biol*, 5(1):33, 2005.

[4] J.R. Brown. Ancient horizontal gene transfer. *Nat Rev Genet*, 4(2):121–132, Feb 2003.

[5] J.H. Camin and R.R. Sokal. A method for reducing branching sequences in phylogeny. *Evolution*, 19:311–326, 1965.

[6] M. Csűrös and I. Miklós. A probabilistic model for gene content evolution with duplication, loss and horizontal transfer. In *In the tenth annual international conference on Research in Computational Molecular Biology (RECOMB)*, pages 206–220. Springer, 2006.

[7] W.F. Doolittle. Phylogenetic classification and the universal tree. *Science*, 284(5423):2124–2129, Jun 1999.

[8] W.F. Doolittle and E. Bapteste. Pattern pluralism and the tree of life hypothesis. *Proc Natl Acad Sci U S A*, 104(7):2043–2049, Feb 2007.

[9] A.J. Drummond, S.Y. Ho, M.J. Phillips, and A. Rambaut. Relaxed phylogenetics and dating with confidence. *PLoS Biol*, 4(5):e88, May 2006.

[10] W. Feller. *An introduction to probability theory and its applications*, volume 1. Wiley, 1968.

[11] J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J Mol Evol*, 17(6):368–376, 1981.

[12] J. Felsenstein. *Inferring phylogenies*. Sinauer Associates, 2003.

[13] A. Gelman and D.B. Rubin. Inference from iterative simulation using multiple sequences. *Stat Sci*, 7(4):457–472, 1992.

[14] J.P. Gogarten, W.F. Doolittle, and J.G. Lawrence. Prokaryotic evolution in light of gene transfer. *Mol Biol Evol*, 19(12):2226–2238, Dec 2002.

[15] M. Goodman, J. Cselusniak, G.W. Moore, A.E. Romero-Herrera, and G. Matsuda. Fitting the gene lineage into its species lineage: A parsimony strategy illustrated by cladograms constructed from globin sequences. *Syst Zool*, 28:132–168, 1979.

[16] F. Griffith. The significance of pneumococcal types. *J Hyg*, 27:113–159, 1928.

[17] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving ordinary differential equations I: nonstiff problems*. Springer-Verlag, 1993.

[18] M. Hallett, J. Lagergren, and A. Tofigh. Simultaneous identification of duplications and lateral transfers. *Proceedings of the eighth annual international conference on Research in Computational Molecular Biology (RECOMB)*, pages 347–356, 2004.

[19] M.T. Hallett and J. Lagergren. Efficient algorithms for lateral gene transfer problems. *Proceedings of the fifth annual international conference on Research in Computational Biology (RECOMB)*, 2001.

[20] J.P. Huelsenbeck, B. Rannala, and B. Larget. A Bayesian framework for the analysis of cospeciation. *Evolution*, 54(2):352–364, Apr 2000.

[21] J.P. Huelsenbeck and F. Ronquist. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, Aug 2001.

[22] D.T. Jones, W.R. Taylor, and J.M. Thornton. The rapid generation of mutation data matrices from protein sequences. *Comput Appl Biosci*, 8(3):275–282, Jun 1992.

[23] P.J. Keeling and J.D. Palmer. Horizontal gene transfer in eukaryotic evolution. *Nat Rev Genet*, 9(8):605–618, Aug 2008.

[24] David G. Kendall. On the generalized "birth-and-death" process. *Ann Math Stat*, 19:1–15, 1948.

[25] H. Kishino, J.L. Thorne, and W.J. Bruno. Performance of a divergence time estimation method under a probabilistic model of rate evolution. *Mol Biol Evol*, 18(3):352–361, Mar 2001.

[26] A.G. Kluge and J.S. Farris. Quantitative phyletics and the evolution of anurans. *Syst Zool*, 18(1):1–32, 1969.

[27] J. Lederberg and E. Tatum. Gene recombination in Escherichia coli. *Nature*, 158:558, October 1946.

[28] J. Lederberg and E.L. Tatum. Novel genotypes in mixed cultures of biochemical mutants of bacteria. In *Cold Spring Harb Symp Quant Biol*, volume 11, pages 113–114, 1946.

[29] T. Lepage, D. Bryant, H. Philippe, and N. Lartillot. A general comparison of relaxed molecular clock models. *Mol Biol Evol*, 24(12):2669–2680, Dec 2007.

[30] E. Lerat, V. Daubin, H. Ochman, and N.A. Moran. Evolutionary origins of genomic repertoires in bacteria. *PLoS Biol*, 3(5):e130, May 2005.

[31] K. Ochiai, T. Yamanaka, K. Kimura, and O. Sawada. Inheritance of drug resistance (and its transfer) between Shigella strains and between Shigella and E. coli strains. *Nihon Iji Shimpo*, 1861:34, 1959.

[32] H. Ochman, J.G. Lawrence, and E.A. Groisman. Lateral gene transfer and the nature of bacterial innovation. *Nature*, 405(6784):299–304, May 2000.

[33] B. Rannala and Z. Yang. Inferring speciation times under an episodic molecular clock. *Syst Biol*, 56(3):453–466, Jun 2007.

[34] J.L. Thorne and H. Kishino. Divergence time and evolutionary rate estimation with multilocus data. *Syst Biol*, 51(5):689–702, Oct 2002.

[35] J.L. Thorne, H. Kishino, and I.S. Painter. Estimating the rate of evolution of the rate of molecular evolution. *Mol Biol Evol*, 15(12):1647–1657, Dec 1998.

[36] C.R. Woese. On the evolution of cells. *Proc Natl Acad Sci U S A*, 99(13):8742–8747, Jun 2002.

[37] Ö. Åkerborg, B. Sennblad, L. Arvestad, and J. Lagergren. Simultaneous Bayesian gene tree reconstruction and reconciliation analysis. *Proc Natl Acad Sci U S A*, 106(14):5714–5719, Apr 2009.

[38] Ö. Åkerborg, B. Sennblad, and J. Lagergren. Birth-death prior on phylogeny and speed dating. *BMC Evol Biol*, 8:77, 2008.

# IV

# A global structural EM algorithm for a model of cancer progression

Ali Tofigh, Erik Sjölund, Mattias Höglund, and Jens Lagergren

**Abstract**

Cancer has complex patterns of progression that include converging as well as diverging progressional pathways. Vogelstein's path model of colon cancer was clearly a pioneering contribution to cancer research. Since then, several attempts have been made at obtaining mathematical models of cancer progression, devising training algorithms, and applying these to cross-sectional data.

Beerenwinkel *et al.* provided, what they coined, EM-like algorithms for Oncogenetic Trees (OTs) and mixtures of such. Given the small size of current and future data sets, it is important to minimize the number of parameters of a model. For this reason, also we focus on tree-based models and introduce Hidden-variable Oncogenetic Trees (HOTs). In contrast to OTs, HOTs allow for errors in the data and thereby provide more realistic modeling. We also design global structural EM algorithms for learning HOTs and mixtures of HOTs (HOT-mixtures). The algorithms are global in the sense that, during the M-step, they find a structure that yields a global maximum of the expected complete log-likelihood rather than merely one that improves it.

The algorithm for single HOTs performs very well on reasonable-sized data sets, while that for HOT-mixtures requires data sets of sizes obtainable only with tomorrows more cost efficient technologies. To facilitate analysis of complex cytogenetic data sets requiring more than one HOT, we devise a decomposition strategy based on Principal Component Analysis and train parameters on a colon cancer data set. The method so obtained is then successfully applied to kidney cancer.

## 1 Introduction

We view cells in cancer progression as progressing towards further malignancy by repeatedly being exposed to genetic or epigenetic aberrations that up-regulate, down-regulate, or dysregulate pathways. Thus, cancer progression is viewed as a walk through a set of states, each representing a set of affected pathways and the type of alterations they have been subjected to. This can be represented by a *a progression graph*, which is a directed graph where the vertices are states and the arcs represent possible transitions between them. Although a tumor is typically heterogeneous with respect to cell types, we make the common assumption that it is homogeneous; a proper discussion of this subject lies outside the scope of this

paper. Consider a situation where it is possible to repeatedly sample from the same tumor of a mouse and identify the malign cell types. Clearly, this would provide a path through our progression graph, and by concatenating paths obtained from different mice having the same cancer type the entire progression graph could conceivably be inferred. In this hypothetical situation, transition probabilities could also be estimated, which would provide a Markov chain. Unfortunately, the accessible biological samples typically do not comprise a time series for each tumor, but are cross-sectional, i.e., a data set is a collection of tumors that each have been removed from a different diseased individual after diagnosis.

In the near future, multiple types of high throughput (HTP) data will be available for large collections of tumors, providing great opportunities for state identification, and thereby, providing computational challenges for progression model inference. In this paper, we focus on cytogenetic data for colon and kidney cancer, mostly due to the availability of cytogenetic data for large numbers of tumors provided by the Mitelman database [17]. Rather than attempting to find a progression graph, we develop a tree-based model, since they have the significant advantage of having fewer parameters. It also turns out that these models allow for efficient algorithms. One of the main motivations for our models and inference methods is that they enable analysis of future HTP-data, which most likely will require the ability to handle large numbers of mutational events.

## 1.1 Mathematical models and algorithms

Vogelstein made a pioneering contribution to cancer research by proposing a path model for colon cancer. Since then, numerous examples of narrative models, often depicted with DAGs, e.g., [20], have been published. In an effort to provide mathematical models of cancer progression, Desper *et al.* [6] introduced the Oncogenetic Tree model where observable variables corresponding to aberrations are associated with vertices of a tree. They then proceeded to show that an algorithm based on Edmonds's optimum branching algorithm will, with high probability, correctly reconstruct an Oncogenetic Tree from sufficiently long series of data generated from it. In [7], another algorithm is described and shown to converge to an Oncogenetic Tree that generates a distribution close to the one generated by the true tree.

The Oncogenetic Tree model suffers from two problems: (1) monotonicity: an aberration associated with a child cannot occur unless the aberration associated with its parent has occurred, and (2) non-convergence: different progression paths cannot converge on the same aberration, as often is the case in tumor progression. In an attempt to remedy these problems, the Network Aberration Model was proposed [12, 18]. However, the computational problems associated with these network models are hard; for instance, no efficient EM algorithm for training is yet known. In another attempt, Beerenwinkel *et al.* used mixtures of Oncogenetic Trees to overcome the problem of non-convergence, but without removing the monotonicity and only obtaining an algorithm with an EM-like structure, which has not been proved to deliver a locally optimal maximum likelihood (ML) solution [1, 2, 19]. These mixture models were originally developed to model HIV evolution and were only later applied to model cancer progression.

It is customary to distinguish between EM algorithms and generalized EM algorithms, the difference being that in the M-step of the former, parameters are found that maximize the expected complete log-likelihood, whereas in the latter, parameters are found that merely improve it. As Friedman notes in his article on the Bayesian Structural EM algorithm [10], the same distinction can be made regarding the maximization over structures. Clearly, it would be convenient to use the same terminology for structural EM algorithms as for ordinary EM algorithms. However, for structural EM algorithms, the distinction is often not made, and even researchers that consider themselves experts in the field seem to be unaware of it. For this reason, we define *global* structural EM algorithms to be EM algorithms that in the M-step find a structure yielding a global maximum of the expected complete log-likelihood (rather than merely improving the expected complete log-likelihood).

In the learning literature, there are several previous results on learning trees and global structural EM algorithms. Chow and Lieu considered trees where the vertices were associated with observable variables and gave an efficient algorithm for finding a globally optimal ML solution [4]. Subsequently, Meila *et al.* presented a global structural EM algorithm for finding the ML mixture of trees [16], as well as MAP solutions with respect to various priors. Friedman *et al.* [11] described a global structural EM algorithm for phylogenetic trees. It is interesting, in relation to the present result, to note that Friedman *et al.* consider phylogenetic trees, i.e., trees with observable variables associated to leaves and hidden variables associated to the internal vertices, and where, moreover, a reversible probabilistic model relates any pair of variables associated with neighboring vertices. Solving the maximum spanning tree problem for a weighted graph is a main component of all these algorithms.

We present the Hidden-variable Oncogenetic Tree (HOT) model where a hidden and an observable variable are associated with each vertex of a rooted directed tree. The value of the hidden variable indicates whether the tumor progression has reached the vertex (a value of one means that cancer progression has reached the vertex and zero that it has not), while the value of the observable variable indicates whether a specific aberration has been detected in HTP-data (a value of one represents detection and zero the opposite). This interpretation provides several relations between the variables in a HOT that are specified in the formal definition of our model. An asymmetric relation is required between the hidden variables associated with the two endpoints of an arc of the directed tree. Because of the asymmetry, the global structural EM algorithm that we derive for the HOT ML problem can, in contrast to the above mentioned algorithms, not be based on a maximum spanning tree algorithm, and is instead based on the optimal branching algorithm [3, 15, 21]. Having so rectified the monotonicity problem, we proceed to obtain a model allowing for a higher degree of convergence by introducing mixtures of HOTs (HOT-mixtures) and, in contrast to Beerenwinkel *et al.*, we derive a proper structural EM algorithm for training these.

We focus on tree models for two reasons: (1) there is a global structural EM algorithm for inference from cross-sectional data and (2) tree models have few parameters. The latter is very important due to the relatively small number of data points available, both in data sets today and in those of the future, compared to the number of mutational events under consideration. It has been observed that cancer progression paths can diverge as well as

converge. In one form of cancer two tumors having different possibly disjoint sets of aberrations such as $\{1, 2\}$ and $\{3, 4\}$ may both obtain the aberration 5, i.e., they converge in the sense that they both obtain the same aberration. It is also possible to have convergence when two tumors with different sets of aberrations both make transitions to the same set of aberrations. Divergence is possible when two different tumors having progressed along the same path of states up to some point in the next step acquire different aberrations. The underlying tree structure of a HOT allows for divergence and convergence, and HOT-mixtures allow for convergence to an even greater extent. Again, in our HOT model, hidden variables model the cancer progression and observable variables correspond to detection of progression in data. So, in contrast to Oncogenetic trees and mixtures of such, HOTs and HOT-mixtures can handle cases where in some tumors a subset of aberrations are undetected in HTP-data.

In Section 2, we show how to model cancer progression by using HOTs and HOT-mixtures. In section 3, this modeling methodology is applied to cytogenetic copy number aberration (CNA) data for colon and kidney cancer.

# 2 HOTs and the novel global structural EM algorithm

This section contains four subsections. In the first, we introduce the HOT model and compare it to the OT model. Subsection 2.2 contains a description of our EM algorithm for training HOTs. In subsection 2.3, we show how to compute certain probabilities that are required during training. Finally, an EM algorithm for training HOT-mixtures is described in subsection 2.4.

## 2.1 Hidden-variable Oncogenetic Trees

We will denote the set of observed data points $D$ and an individual data point $X$. In Section 3, we will apply our methods to CNA, i.e., a data point will be a set of observed CNA, but in general, more complex events can be used.

A *rooted directed tree* $T$ consists of a set of vertices, denoted $V(T)$ and a set of arcs denoted $A(T)$. An arc $\langle u, v\rangle$ is directed from the vertex $u$ called its *tail* towards the vertex $v$ called its *head*. If there is an arc with tail $p$ and head $u$ in a directed tree $T$, then $p$ is called the parent of $u$ in $T$ and denoted $p(u)$ (the tree $T$ will be clear from context).

An OT is a rooted directed tree where there is an aberration associated with each vertex and a probability associated with each arc. One can view an OT as generating a set of aberrations by first visiting the root and then continuing towards the leaves (preorder) visiting each vertex with the probability of its incoming arc if the parent has been visited, and with probability zero if the parent has not been visited. Finally, the result of the progression is the set of aberrations associated with the visited vertices.

In Figure 1(b), an OT for CNA is depicted. It can generate the following sets of CNAs: $\emptyset$, $\{-3p\}$, $\{-3p, -4p\}$, $\{-3p, +Xp\}$, $\{-3p, -4p, +Xp\}$, $\{+17q\}$, $\{-3p, +17q\}$, $\{-3p, -4p, +17q\}$, $\{-3p, +Xp, +17q\}$, and $\{-3p, -4p, +Xp, +17q\}$ (all these aberrations are written in the standard notation for CNAs in cytogenetic data, i.e., each represents a duplication (+) or

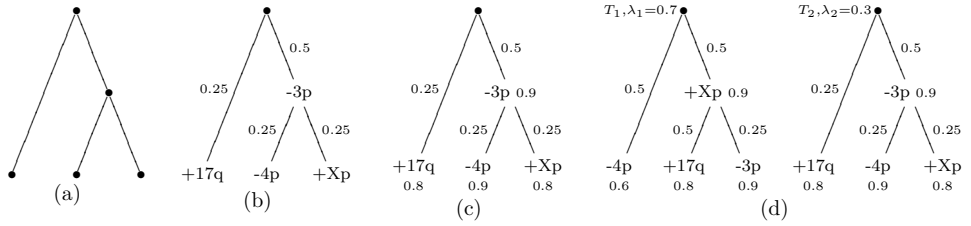



Figure 1: **(a)** *A rooted directed tree with the root at the top. All arcs are directed downwards, i.e., away from the root.* **(b)** *An OT with probabilities associated with arcs and CNAs associated with vertices.* **(c)** *A HOT with probabilities associated with arcs (indicating the probability that the hidden variable associated with the head of the arc receives the value* 1 *conditioned that the hidden variable associated with the tail has this value), and CNAs as well as probabilities associated with vertices (indicating the probability that the observable variable associated with the vertex receives the value* 1 *conditioned that the hidden variable associated with the vertex has received this value).* **(d)** *A HOT-mixture consisting of two HOTs. The mixing probability for* $T_1$ *is* 0.7 *and that for* $T_2$ *is* 0.3. *So with probability* 0.7 *a synthetic tumor is generated from* $T_1$ *and otherwise one is generated from* $T_2$.

deletion (-) of a specific chromosomal region). Notice that an aberration associated with a vertex cannot occur unless the aberration associated with its parent has occurred. For instance, the set $\{+Xp, +17q\}$ cannot be generated by the OT in Figure 1(b). In a data-modeling context, this is highly undesirable, since data is typically noisy and whatever mutational events we are modeling some of those that have occurred are likely to have gone undetected. Our HOT model does not suffer from this problem.

A Hidden-variable Oncogenetic Tree (HOT) is a directed tree where there is an aberration associated with each vertex and a probability associated with each arc, exactly as in a OT. Moreover, in contrast to the OT, there is also a probability associated with each vertex. One can view the HOT as generating data by first allowing cancer progression to reach a subset of the vertices of the tree, exactly as in an OT, i.e., based on the probabilities associated with the arcs. In the HOT, however, an aberration associated with a vertex reached by the progression process is not automatically generated, instead it is generated with the probability associated with that vertex. As an example consider the HOT illustrated in Figure 1(c). The probability that it generates the set $\{+Xp, +17q\}$ is $0.25{\cdot}0.8{\cdot}0.5{\cdot}0.1{\cdot}0.25{\cdot}0.8{\cdot}(0.75{+}0.25{\cdot}0.1) = 0.00155$.

We will now give a formal definition of HOTs. Notice that the probabilities associated with edges in the description above are the conditional probabilities in (4) and those associated with vertices are the conditional probabilities in (5). A Hidden-variable Oncogenetic Tree (HOT) is a pair $\mathcal{T} = (T, \Theta)$ where:

1. $T$ is a rooted directed tree and $\Theta$ consists of two conditional probability distributions (CPDs), $\theta_X(u)$ and $\theta_Z(u)$, for each vertex $u$;

2. two random variables are associated with each vertex $u \in V(T)$: an observable variable $X(u)$ and a hidden variable $Z(u)$, each assuming the values 0 or 1,

3. the hidden variable associated with the root, $Z(r)$, always assumes the value 1,

4. for each non-root vertex $u$ of $V(T)$, $\theta_Z(u)$ is a conditional probability distribution on $Z(u)$ conditioned by $Z(p(u))$ satisfying $\Pr[Z(u) = 1|Z(p(u)) = 0] = 0$, and

5. for each non-root vertex $u$ of $V(T)$, $\theta_X(u)$ is a conditional probability distribution on $X(u)$ conditioned by $Z(u)$ satisfying $\Pr[X(u) = 1|Z(u) = 0] = 0$.

For practical reasons, in the implementation of the algorithm, we use the condition $\Pr[Z(u) = 1|Z(p(u)) = 0] = \epsilon_Z$, where $\epsilon_z$ is a small value, rather than the strict condition in (4). The motivation is basically the same as for using so called pseudo-counts [8]. Namely, once a parameter receives the value 0 in an EM algorithm for training, it will subsequently not be changed. For modeling reasons, we use the condition $\Pr[X(u) = 1|Z(u) = 0] = \epsilon_X$ for some small $\epsilon_X$ rather than the condition stated in (5).

In Subsection 3, when modeling a collection of tumors represented by CNAs, we will number the CNAs $1, \ldots, n$ and also use these numbers to represent the non-root vertices, and we will consider a CNA $i$ to have happened if and only if $X(i) = 1$, i.e., the final set of aberrations generated is $\{i : X(i) = 1\}$.

It is also possible to have CPDs where $X(u)$ and $Z(u)$ depend on both $X(p(u))$ and $Z(p(u))$ and even to let $X(u)$ depend on all three of $Z(u)$, $X(p(u))$, and $Z(p(u))$. We do not cover these cases in the following text, but our arguments can easily be extended to also cover these.

## 2.2 The novel global structural EM algorithm for HOTs

When viewing probabilistic models as generating data, the model-training problem can be cast as an optimization problem where the goal is to find the maximum likelihood solution, i.e., the model that with the highest probability generates the observed data. This optimization problem is often solved using an Expectation Maximization (EM) algorithm [5], which is not guaranteed to deliver a globally optimal solution but used to obtain locally optimal ones.

The EM theory shows that given a current solution, another solution with higher likelihood can be found by maximizing the so-called expected complete log-likelihood or the $Q$-term. Friedman *et al.* [11] extended the use of EM algorithms from the standard parameter estimation to also finding an optimal structure. In their case, the probabilistic model was reversible which makes it possible to maximize the expected complete likelihood over all trees, using a maximum spanning tree algorithm. In our case, the pair-wise relations between hidden variables are asymmetric. However, as shown below, the maximization of the expected complete log-likelihood can in our case be solved using Edmonds's optimal branching algorithm. Tarjan's variation of Edmonds's algorithm runs in quadratic time [3, 15, 21].

The *weighted* expected complete log-likelihood function will be useful when treating HOT-mixtures. We introduce it already here and also show how to maximize it. The expected complete log-likelihood of a HOT $\mathcal{T}'$ with respect to another HOT $\mathcal{T}$, weighted by a function $f$, and with our observed variables as parameters, is defined as

$$Q_f(\mathcal{T}';\mathcal{T}) = \sum_{X\in D}\sum_{Z} f(X)\Pr[Z|X,\mathcal{T}]\log\Pr[Z,X|\mathcal{T}']. \tag{1}$$

We now show that if $f$ can be evaluated in constant time, then the HOT $\mathcal{T}'$ that maximizes (1) can be found in time $O(n^2)$, where $n$ is the number of aberrations or vertices.

Following the standard derivation of an EM algorithm, it can be shown that $Q_f(\mathcal{T}';\mathcal{T})$ equals

$$\sum_{\langle u,v\rangle\in A(T')}\sum_{a,b\in\{0,1\}}\sum_{X\in D} f(X)\Pr[Z(v)=a,Z(u)=b|X,\mathcal{T}]\log\Pr[Z(v)=a|Z(u)=b,\theta'_Z(u)]$$
$$+\sum_{\langle u,v\rangle\in A(T)}\sum_{\sigma,a\in\{0,1\}}\sum_{X\in D:X(u)=\sigma} f(X)\Pr[Z(v)=a|X,\mathcal{T}]\log\Pr[X(v)=\sigma|Z(v)=a,\theta'_X(u)].$$

As long as the directed tree $T'$ is fixed, the standard EM methodology (see for instance [8]) can be used to find the $\Theta'$ that maximizes $Q_f(T',\Theta';\mathcal{T})$, as follows. First, let

$$A_u(a,b) = \sum_{X\in D} f(X)\Pr[Z(u)=a,Z(p'(u))=b|X,\mathcal{T}] \tag{2}$$

and

$$B_u(\sigma,a) = \sum_{X\in D:X(u)=\sigma} f(X)\Pr[Z(u)=a|X,\mathcal{T}]. \tag{3}$$

Then the $\Theta'$ that for a fixed $T'$ maximizes $Q_f(\mathcal{T}';\mathcal{T})$ (i.e. $Q_f(T',\Theta';\mathcal{T})$) is given by letting

$$\Pr[Z(u)=a|Z(p'(u))=b,\theta'_Z(u)] = A_u(a,b)/(\sum_{a\in\{0,1\}} A_u(a,b))$$

and

$$\Pr[X(u)=\sigma|Z(u)=a,\theta'_Z(u)] = B_u(\sigma,a)/(\sum_{\sigma\in\{0,1\}} B_u(\sigma,a)).$$

In the next Section 2.3, we will describe how the probabilities on the right hand sides of (2) and (3) can be computed. The time required for computing all such probabilities will turn out to be no more than $O(|D|\cdot n^2)$, where $n$ is the number of aberrations.

For each arc $\langle p,u\rangle$ of $T'$, using the CPDs defined above, we define the weight of the arc, specific to this tree to be

$$\sum_{a,b\in\{0,1\}}\sum_{X\in D} f(X)\Pr[Z(u)=a,Z(p'(u))=b|X,\mathcal{T}]\log\Pr[Z(u)=a|Z(p'(u))=b,\theta'_Z(u)]$$
$$+\sum_{b\in\{0,1\}}\sum_{X\in D} f(X)\Pr[Z(u)=a|X,\mathcal{T}]\log\Pr[X(u)|Z(u)=a,\theta'_X(u)].$$

We now make two important observations from which it follows how to maximize the weighted expected complete log-likelihood over all directed trees. First, notice that if two directed trees $T'$ and $T''$ have a common arc $\langle p, u\rangle$, then this arc has the same weight in these two trees. This allows us to define a complete directed and arc-weighted graph $D$ (i.e., a combinatorial structure with a set of vertices and an arc in each direction between any two vertices) with the same vertex set as the tree $T$, and define the weight of the arc $\langle u, v\rangle$ in this directed graph to be the same as in any directed tree containing the arc.

An optimal arborescence of a directed graph is a rooted directed tree on the same set of vertices as the directed graph that has exactly one directed path from one specific vertex called the root to any other vertex and, moreover, has maximum arc weight sum among all such rooted directed trees. Now we are in position to conclude that Edmonds's optimal branching algorithm (of which a variation can produce an optimal arborescence) can be used to maximize the weighted expected complete log-likelihood. For any branching $T'$ of $D$, the sum of the weights of its arcs equals by construction the maximum value of $Q_f(T', \Theta'; \mathcal{T})$ for any $\Theta'$. From this follows that, a (spanning) directed tree $T'$ is an optimal branching of $D$ if and only if $T'$ maximize the $Q_f$ term. So applying Tarjan's variation of Edmonds's algorithm [3, 15, 21] to $D$ gives the desired directed tree. In the next subsection, we show how to compute the probabilities required in (2) and (3), thereby, complete the description of our model-training algorithm for HOTs.

## 2.3 Computing the required probabilities

The most basic computation for a HOT $\mathcal{T} = (T, \Theta)$ is computing the probability that an observation $X$ is generated from $\mathcal{T}$, i.e., $\Pr[X|\mathcal{T}]$. This probability as well as the probabilities $\Pr[Z(u) = a, X|\mathcal{T}]$ and $\Pr[Z(u) = a, Z(p(u)) = b, X|\mathcal{T}]$ can be computed in linear time using dynamic programming, i.e., a procedure very similar to the pruning algorithm used to compute likelihoods of phylogenetic trees [9]. Doing so for all vertices $u$ can, hence, be done in time $O(n^2)$. Using the above probabilities, we can in linear time compute $\Pr[Z(u) = a|X, \mathcal{T}]$ and $\Pr[Z(u) = a, Z(p(u)) = b|X, \mathcal{T}]$ for all vertices $u$. Finally, using the so computed probabilities, the probability $\Pr[Z(u) = a, Z(v) = b|X, \mathcal{T}]$ can then be computed using techniques analogous to those appearing in [11].

## 2.4 HOT-mixtures

In the previous section, we considered a HOT $\mathcal{T} = (T, \Theta)$. We will now extend the model to HOT-mixtures by including an initial random choice of one out of several HOTs and letting the final outcome be generated by the chosen HOT. We will also obtain an EM based model-training algorithm for HOT-mixtures by showing how to optimize expected complete log-likelihoods for HOT-mixtures. Formally, we will use $k$ HOTs $\mathcal{T}_1, \ldots, \mathcal{T}_k$ and a random mixing variable $I$ taking on values in $1, \ldots, k$. The probability that $I$ gets the value $i$ is denoted $\lambda_i$ and $\lambda = (\lambda_1, \ldots, \lambda_k)$ is a vector of parameters of the model, in addition to those

of the HOTs ($\lambda_1, \ldots, \lambda_k$ are constrained to sum to 1). The following notation is convenient

$$\gamma_i(X) = \Pr[I = i|X, M] = \frac{\lambda_i \Pr[X|\mathcal{T}_i]}{\sum_{j\in[k]} \lambda_j \Pr[X|\mathcal{T}_j]}.$$

For a HOT-mixture, the expected complete log-likelihood can be expressed as follows

$$\sum_{X\in D} \sum_{Z,I} \Pr[Z, I|X, M] \log \Pr[Z, I, X|M']. \tag{4}$$

Using standard EM methodology, it is possible to show that (4) can be maximized by independently maximizing

$$\sum_{i\in[k]} \sum_{X\in D} \gamma_i(X) \log(\lambda'_i) \tag{5}$$

and, for each $i = 1, \ldots, k$, maximizing

$$\sum_{X\in D} \sum_{Z} \Pr[Z|X, \mathcal{T}_i]\gamma_i(X) \log(\Pr[Z, X|\mathcal{T}'_i]) \tag{6}$$

Finding a $\lambda' = \lambda'_1, \ldots, \lambda'_k$ maximizing (5) is straightforward (see for instance [8]) and, for each $i = 1, \ldots, k$, finding a $\mathcal{T}'_i$ maximizing (6) can be done as described in the previous subsections.

# 3 Results

In this section, we report results obtained by applying our algorithms to synthetic data as well as cytogenetic cancer data. For ease of notation, we will denote the parameters of the model as follows:

$$\begin{aligned}
p_z(u) &= \Pr[Z(u) = 1|Z(p(u)) = 1] \\
p_x(u) &= \Pr[X(u) = 1|Z(u) = 1] \\
e_z(u) &= \Pr[Z(u) = 1|Z(p(u)) = 0] \\
e_x(u) &= \Pr[X(u) = 1|Z(u) = 0].
\end{aligned}$$

We will collectively call the three parameters, $(1 - p_x)$, $e_z$, and $e_x$, the *error parameters*.

In the standard version of the EM algorithm, four parameters are associated with each edge of a HOT. In order to reduce the total number of parameters, it is possible to let some of the four parameters be global instead. For example, in the case of $e_x$, this means that we would require that $e_x(u) = e_x(u')$ for all pairs of vertices $u$ and $u'$. Ideally, we would like to let all the error parameters be global. However, for technical reasons, requiring that $e_z$ be global makes it impossible to derive an EM algorithm. Therefore, we will distinguish between two different versions of the algorithm: one with *free parameters* and one with *global*

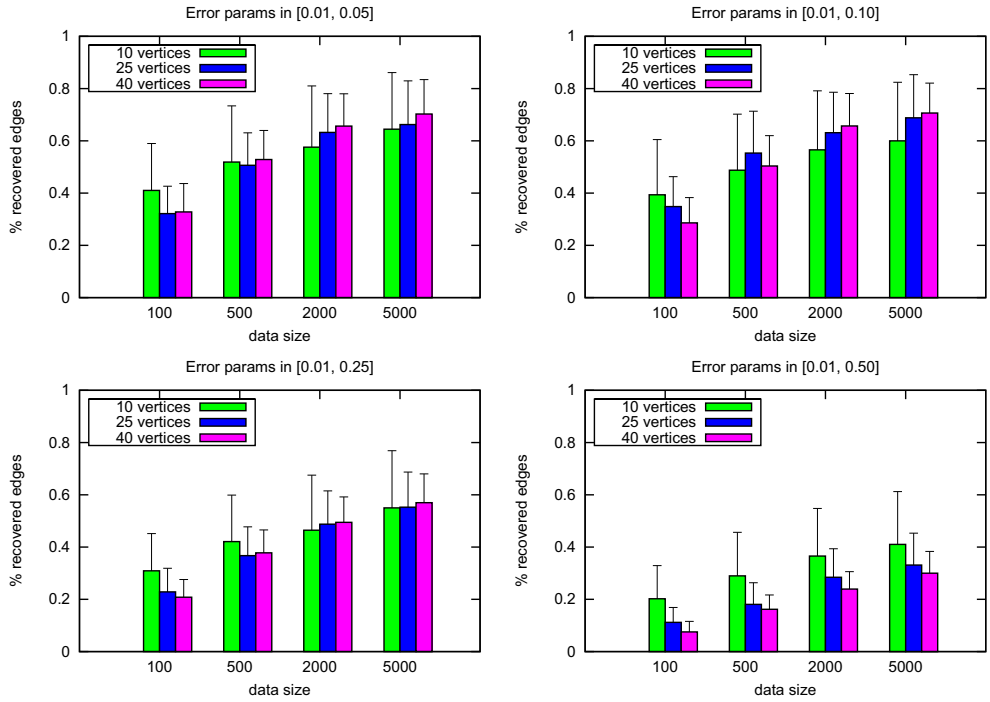



Figure 2: Histograms showing the mean percentage of edges that were correctly recovered by the algorithm for the free parameter case together with errorbars showing one standard deviation.

*parameters*. The free parameter version then corresponds to the standard EM algorithm, while the global parameter version corresponds to letting $(1 - p_x)$ and $e_x$ be global. When evaluating the global parameter version of the algorithm using synthetic data, we will follow the convention of letting all three error parameters be global when generating data.

Other conventions used for all the tests described here include the following. Unless stated otherwise, we enforce an upper limit of 0.5 on $e_z$ and $e_x$. Also, when running the algorithm on a dataset, we first run the algorithm on a set of randomly generated start HOTs or start HOT-mixtures for 10 iterations. The HOT or HOT-mixture that resulted in the best likelihood is then run until convergence. Unless stated otherwise, the number of start trees or mixtures is 100.

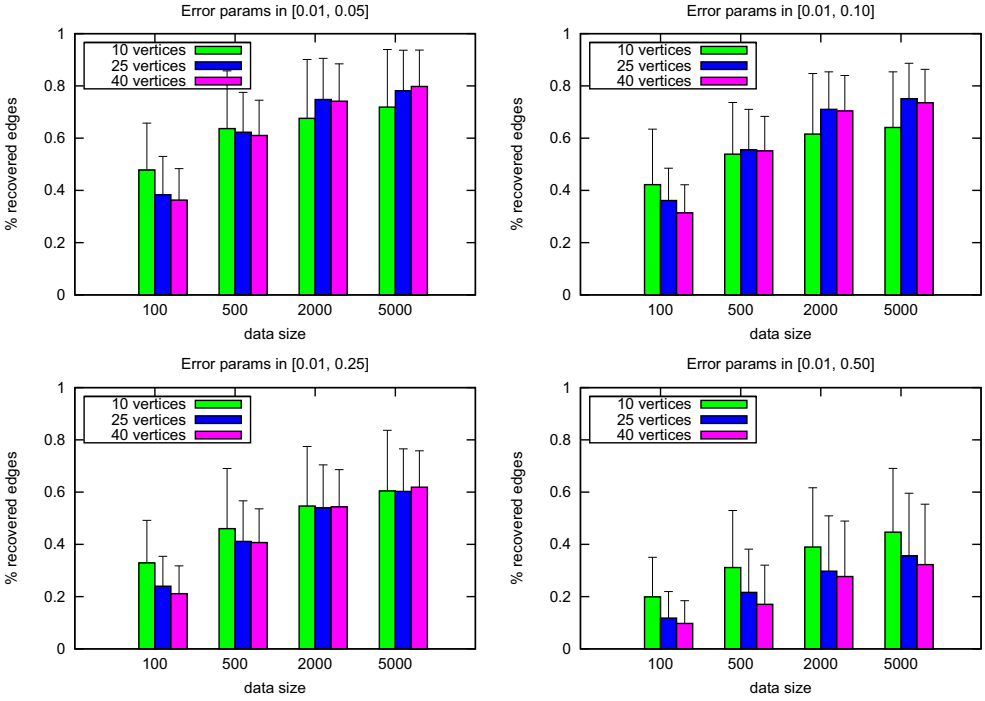



Figure 3: Histograms showing the mean percentage of edges that were correctly recovered by the algorithm for the global parameter case together with errorbars showing one standard deviation.

## 3.1 Tests on Synthetic Datasets

### 3.1.1 Single HOTs

In order to test the algorithm's ability to recover a HOT from data, we generated random HOTs with different sizes and parameters. We then generated data from these HOTs and attempted to recover the HOTs using the hotmix algorithm. The sizes of the HOTs were fixed at 10, 25, and 40 vertices. The parameters on the edges, i.e., the probabilities $p_z$, $p_x$, $e_z$, and $e_x$, were chosen uniformly in the intervals

$$p_z \in [0.1, 1.0], \tag{7}$$

$$(1 - p_x), e_x, e_z \in [0.01, q], \tag{8}$$

where $q \in \{0.05, 0.10, 0.25, 0.50\}$. For each combination of possible sizes and values for $q$, 100 HOTs were generated for a total of $3 \times 4 \times 100 = 1200$ HOTs. Data was generated from each HOT with 100, 500, 2000, and 5000 datapoints. Each dataset was then passed to the algorithm and the resulting HOT was compared to the original HOT. The result of this comparison can be seen in Figure 2. An edge of the original HOT connecting one specific aberration to another is considered to have been correctly recovered if the HOT obtained from the algorithm connects the same two aberrations in the same direction.

We also tested how the reduction of parameters affected the results by generating HOTs with global error parameters. This is the so-called "global parameters" case as described in the introduction. We then applied the relevant version of the algorithm, and the results can be seen in Figure 3. As shown by the figures, there is a slight improvement when the number of parameters is reduced.

We also compared the performance of our algorithms with that of *Mtreemix* by Beerenwinkel *et al* [2]. The generated data from our single HOTs were passed to Mtreemix and the same criteria as above were used to detect correctly recovered edges (no special options were set when running Mtreemix on data generated with global parameters since no distinction between global and free parameters can be made on oncogenetic trees) Figure 4 and 5 show the results. As can be seen, Mtreemix outperforms our methods when the HOTs and the error parameters are small, and our algorithms outperform Mtreemix significantly as the HOTs or error parameters become larger.

### 3.1.2 HOT Mixtures

We also tested the ability of the algorithm to recover a mixture of two HOTs. The sizes of the HOTs were set at either 10 or 25 vertices (i.e. a total of 18 or 48 edges). The error parameters, which were global, were chosen randomly from a uniform distribution on the interval $[0.01, q]$ where $q \in \{0.05, 0.10, 0.25\}$. Three different mixture distributions on the HOTs were also tested.

When measuring the number of correctly recovered edges, the following procedure was used. Each HOT produced from the algorithm was compared to each HOT from which the data was generated, and the number of correctly recovered edges was noted. The best way

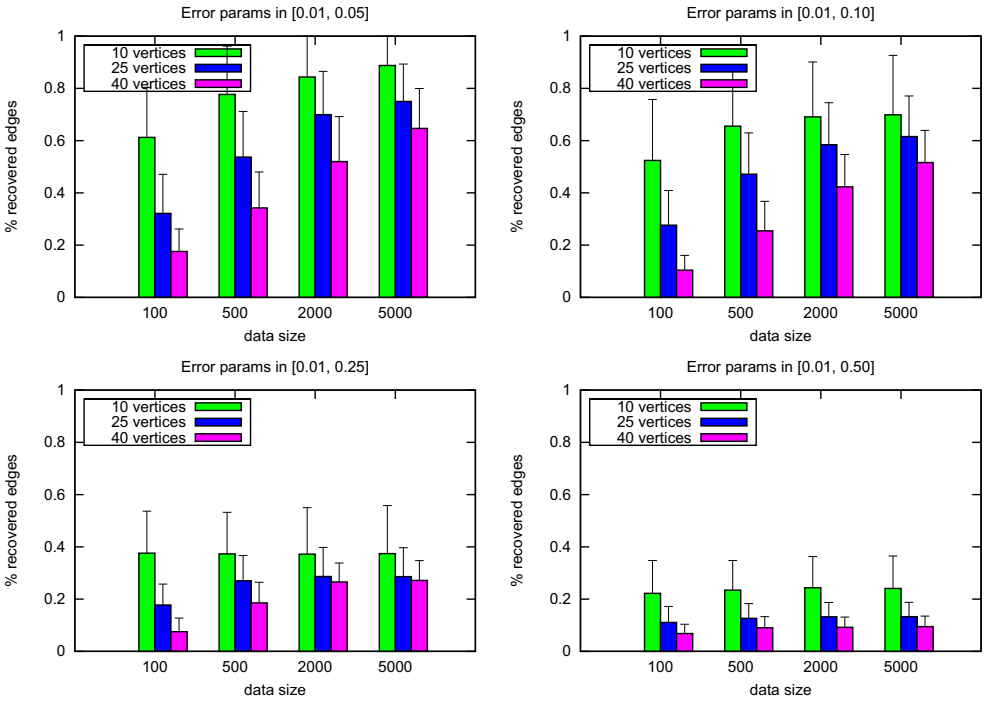



Figure 4: Histograms showing the mean percentage of edges that were correctly recovered by Mtreemix together with errorbars showing one standard deviation. The data was the same as those used in Figure 2.

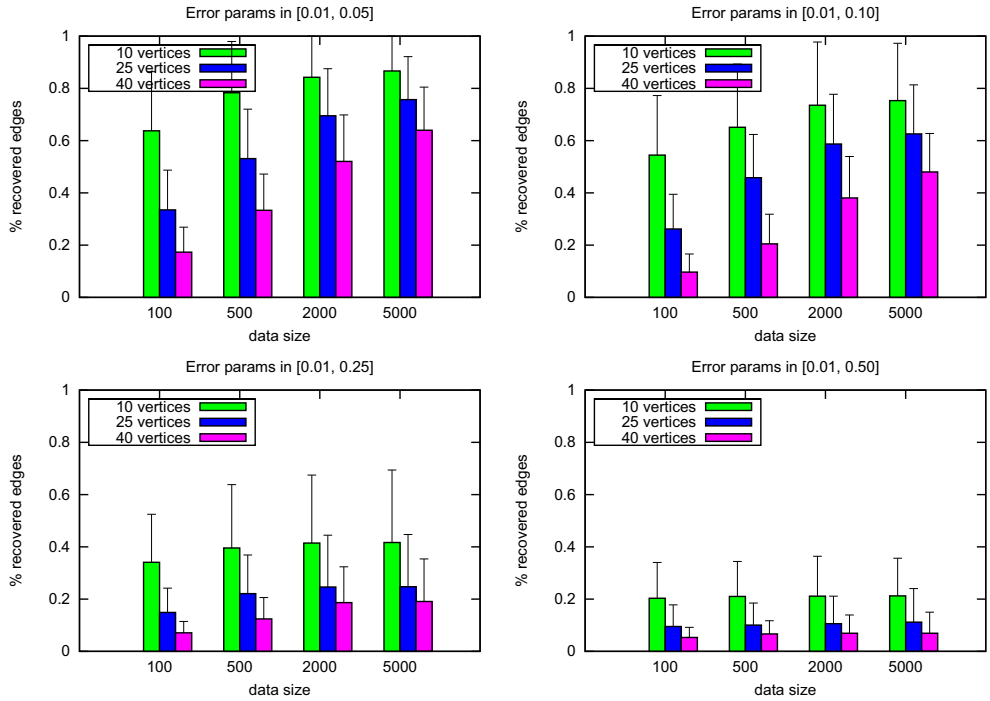



Figure 5: Histograms showing the mean percentage of edges that were correctly recovered by Mtreemix together with errorbars showing one standard deviation. The data was the same as those used in Figure 3.

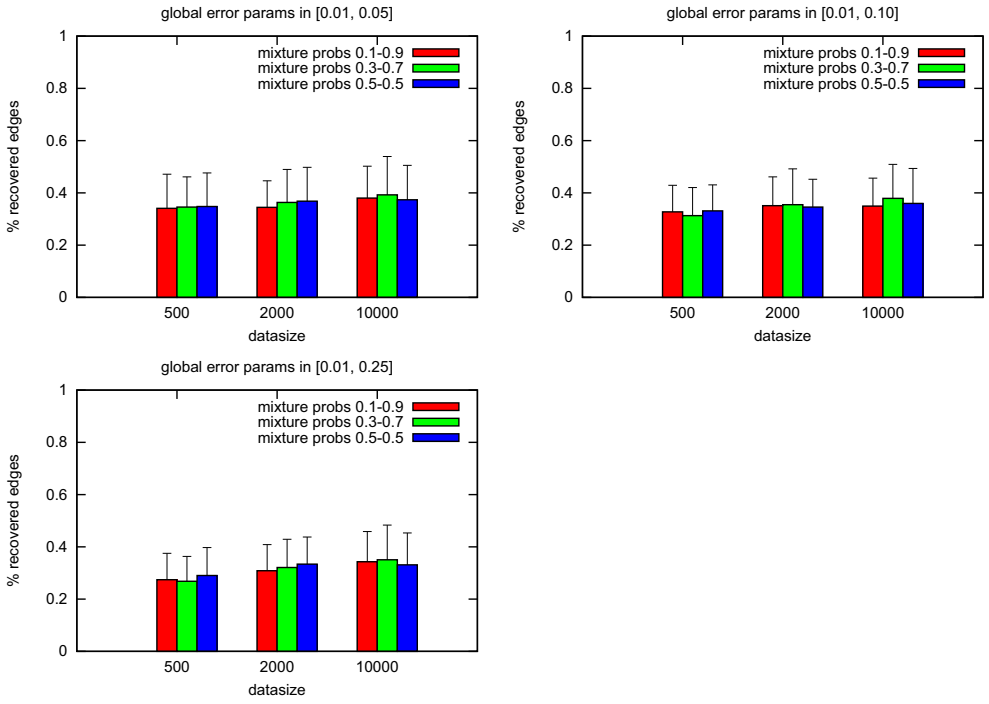



Figure 6: Histograms showing the mean percentage of edges that were correctly recovered for mixtures of two HOTs with 10 vertices each. The errorbars indicate one standard deviation. Each bar represents 100 mixtures.

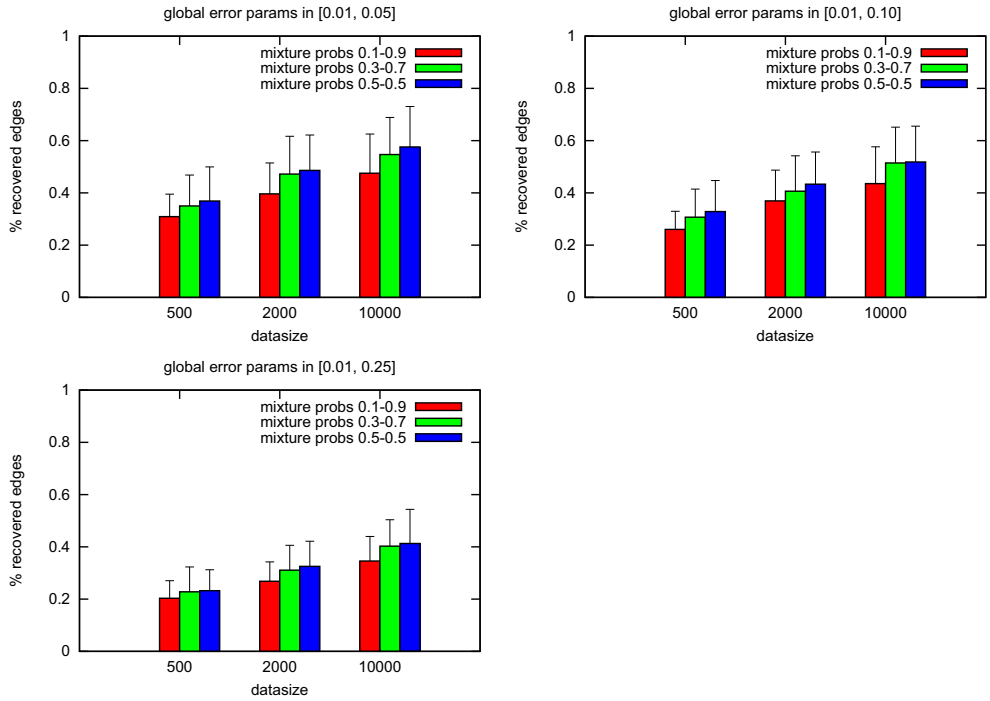



Figure 7: Histograms showing the mean percentage of edges that were correctly recovered for mixtures of two HOTs with 25 vertices each. The errorbars indicate one standard deviation. Each bar represents 100 mixtures.

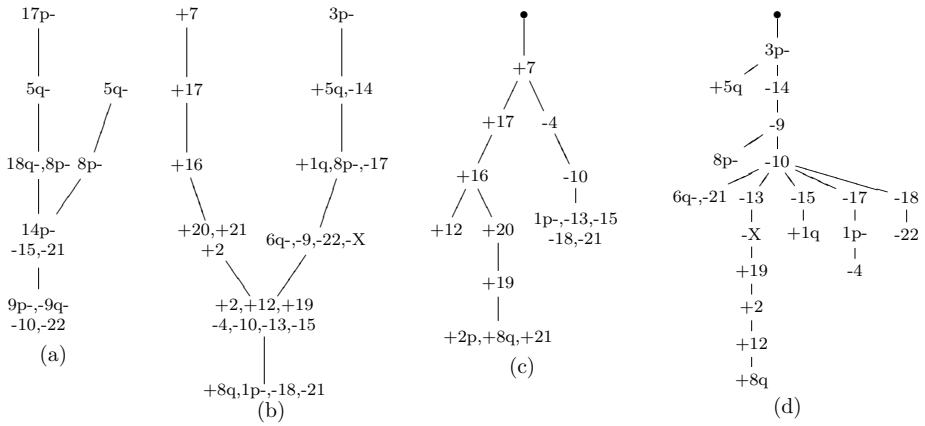



Figure 8: **HOTs obtained from RCC data.** *(a) shows an adapted version of the pathways for CC data published in [13]. (b) is a figure adapted from [14] showing the pathways obtained from statistical analysis of RCC data. (c) and (d) are the HOTs we obtained from the RCC data using only aberrations on the left and right pathways in (b), respectively.*

of matching the two HOTs produced from the algorithm with the two original HOTs was then determined. The result can be seen in Figure 6 and 7.

For the case with 25 vertices, two features can clearly be distinguished: the results improve as the size of the data increases, and the algorithm performs better when the HOTs have equal probability in the mixture.

## 3.2 Tests on Cancer Data

Our cytogenetic data for colon (CC) and kidney (RCC) cancer consist of 512 and 998 tumors, respectively. The data consist of measurements on 41 common aberrations (18 gains, 23 losses) for CC and 28 (13 gains, 15 losses) for RCC. The data have previously been analyzed in [13] and [14] resulting in suggested pathways of progression. These analyses were based on Principal Component Analysis (PCA) performed on correlations between aberrations and a statistical measure called *time of occurrence* (TO) that is a measure on how early or late an aberration occurs during progression. The aberrations were then clustered based on the PCA and each cluster was manually formed into a pathway (based on PCA and TO). One advantage of our approach is that we are able to replace the manual curation by automated computational steps. Another advantage is that our models assign probabilities to data and the different models can therefore be compared objectively.

We expect the parameters $e_z(u) = \Pr[Z(u) = 1|Z(p(u)) = 0]$ and $e_x(u) = \Pr[X(u) = 1|Z(u) = 0]$ to be small in real data. We obtained the $n$ most correlated aberrations in our CC data, for $n \in \{4, \ldots, 11\}$, and tested different upper limits on $e_z$ and $e_x$. The best

correspondence to previously published analyses of the data was found when $e_z(u) \leq 0.25$ and $e_x(u) \leq 0.01$ with the number of *bad edges* given in the table below. A bad edge is one that contradicts the partial ordering given by the pathways described in [13], of which the relevant part is shown in the Figure 8(a).

| size | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|
| bad edges | 0 | 0 | 0 | 0 | 2 | 2 | 3 | 2 |

Having found upper limits that work well on the CC data, we applied the algorithm with these upper bounds to the RCC data. The earlier analyses in [14] strongly suggests that two HOTs are required to model the RCC data. Given that our mixture model, from synthetic data tests, appears to require substantially more data points to recover the underlying HOTs in a satisfactory manner, we used the results of the analysis in [14] to divide the aberrations into two (overlapping) clusters for which we created HOTs separately. These HOTs can be seen in Figure 8(c) and 8(d) and they show very good agreement to the pathways from [14] shown in Figure 8(b). For instance, each root-to-leaf path in the HOT of Figure 8(c) agrees perfectly with the pathway shown in Figure 8(b).

# References


[1] N. Beerenwinkel, J. Rahnenfuhrer, M. Daumer, D. Hoffmann, R. Kaiser, J. Selbig, and T. Lengauer. Learning multiple evolutionary pathways from cross-sectional data. *J Comput Biol*, 12(6):584–598, Jul 2005.

[2] N. Beerenwinkel, J. Rahnenfuhrer, R. Kaiser, D. Hoffmann, J. Selbig, and T. Lengauer. Mtreemix: a software package for learning and using mixture models of mutagenetic trees. *Bioinformatics*, 21(9):2106–2107, 2005.

[3] P. Camerini, L. Fratta, and F. Maffioli. The k best spanning arborescences of a network. *Networks*, 10(2):91–110, 1980.

[4] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 14(3):462–467, 1968.

[5] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[6] R. Desper, F. Jiang, O.P. Kallioniemi, H. Moch, C.H. Papadimitriou, and A.A. Schaffer. Inferring tree models for oncogenesis from comparative genome hybridization data. *J Comput Biol*, 6(1):37–51, 1999.

[7] R. Desper, F. Jiang, O.P. Kallioniemi, H. Moch, C.H. Papadimitriou, and A.A. Schaffer. Distance-based reconstruction of tree models for oncogenesis. *J Comput Biol*, 7(6):789–803, 2000.

[8] R. Durbin, S.R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis*. Cambridge University Press, Cambridge, 1998.

[9] J. Felsenstein. *Inferring phylogenies*. Sinauer Associates Sunderland, Mass., USA, 2004.

[10] N. Friedman. The bayesian structural em algorithm. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 129–138. Morgan Kaufmann, 1998.

[11] N. Friedman, M. Ninio, I. Pe'er, and T. Pupko. A structural EM algorithm for phylogenetic inference. *J Comput Biol*, 9(2):331–353, 2002.

[12] M. Hjelm, M. Höglund, and J. Lagergren. New probabilistic network models and algorithms for oncogenesis. *J Comput Biol*, 13(4):853–865, May 2006.

[13] M. Höglund, D. Gisselsson, G.B. Hansen, T. Säll, F. Mitelman, and M. Nilbert. Dissecting karyotypic patterns in colorectal tumors: Two distinct but overlapping pathways in the adenoma-carcinoma transition. *Cancer Res*, 62:5939–5946, 2002.

[14] M. Höglund, D. Gisselsson, M. Soller, G.B. Hansen, P. Elfving, and F. Mitelman. Dissecting karyotypic patterns in renal cell carcinoma: an analysis of the accumulated cytogenetic data. *Cancer Genetics and Cytogenetics*, 153(1):1–9, 2004.

[15] R.M. Karp. A simple derivation of edmond's algorithm for optimum branching. *Networks*, 1(265-272):5, 1971.

[16] M. Meila and M.I. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1(1):1–48, 2000.

[17] F. Mitelman, B. Johansson, and F. Mertens. Mitelman database of chromosome aberrations in cancer, 2004. http://cgap.nci.nih.gov/Chromosomes/Mitelman.

[18] M.D. Radmacher, R. Simon, R. Desper, R. Taetle, A.A. Schaffer, and M.A. Nelson. Graph models of oncogenesis with an application to melanoma. *J Theor Biol*, 212(4):535–48, Oct 2001.

[19] J. Rahnenfuhrer, N. Beerenwinkel, W.A. Schulz, C. Hartmann, A. von Deimling, B. Wullich, and T. Lengauer. Estimating cancer survival and clinical outcome based on genetic tumor progression scores. *Bioinformatics*, 21(10):2438–2446, May 2005.

[20] P.T. Simpson, J.S. Reis-Filho, T. Gale, and S.R. Lakhani. Molecular evolution of breast cancer. *J Pathol*, 205(2):248–254, Jan 2005.

[21] R.E. Tarjan. Finding optimum branchings. *Networks*, 7(1):25–36, 1977.