The Orthology Road

Theory and Methods in Orthology Analysis

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

DISSERTATION

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr. rer. nat.)

im Fachgebiet
Informatik

Vorgelegt
von B.Sc. Maribel Hernandez Rosales
geboren am 11. Juni 1977 in Mexiko Stadt, Mexiko

Die Annhame der Dissertation wurde empfohlen von:
1. Prof. Dr. Peter F. Stadler
2. Prof. Dr. David Sankoff

Die Verleihung des akademischen Grades erfolgt mit Bestehen der
Verteidigung am 6. September 2013 mit dem Gesamtprädikat ***magna cum laude***

Contents

Abstract

The evolution of biological species depends on changes in genes. Among these changes are the gradual accumulation of DNA mutations, insertions and deletions, duplication of genes, movements of genes within and between chromosomes, gene losses and gene transfer. As two populations of the same species evolve independently, they will eventually become reproductively isolated and become two distinct species. The evolutionary history of a set of related species through the repeated occurrence of this speciation process can be represented as a tree-like structure, called a phylogenetic tree or a *species tree*. Since duplicated genes in a single species also independently accumulate point mutations, insertions and deletions, they drift apart in composition in the same way as genes in two related species. The divergence of all the genes descended from a single gene in an ancestral species can also be represented as a tree, a *gene tree* that takes into account both speciation and duplication events.

In order to reconstruct the evolutionary history from the study of extant species, we use sets of similar genes, with relatively high degree of DNA similarity and usually with some functional resemblance, that appear to have been derived from a common ancestor. The degree of similarity among different instances of the "same gene" in different species can be used to explore their evolutionary history via the reconstruction of gene family histories, namely gene trees.

Orthology refers specifically to the relationship between two genes that arose by a speciation event, recent or remote, rather than duplication. Comparing orthologous genes is essential to the correct reconstruction of species trees, so that detecting and identifying orthologous genes is an important problem, and a longstanding challenge, in comparative and evolutionary genomics as well as phylogenetics.

A variety of orthology detection methods have been devised in recent years. Although many of these methods are dependent on generating gene and/or species trees, it has been shown that orthology can be estimated at acceptable levels of accuracy without having to infer gene trees and/or reconciling gene trees with species trees.

Therefore, there is good reason to look at the connection of trees and orthology from a different angle: *How much information about the gene tree, the species tree, and their reconciliation is already contained in the orthology relation among genes?*

Intriguingly, a solution to the first part of this question has already been given by Böcker and Dress [Böcker and Dress, 1998] in a different context. In particular, they completely characterized certain maps which they called *symbolic ultrametrics*. Semple and Steel [Semple and Steel, 2003] then presented an algorithm that can be used to reconstruct a phylogenetic tree from any given symbolic ultrametric. In this thesis we investigate a new characterization of orthology relations, based on symbolic ultramterics for recovering the gene tree.

According to Fitch's definition [Fitch, 2000], two genes are (co-)orthologous if their last common ancestor in the gene tree represents a speciation event. On the other hand, when their last common ancestor is a duplication event, the genes are paralogs. The orthology relation on a set of genes is therefore determined by the gene tree and an "event labeling" that identifies each interior vertex of that tree as either a duplication or a speciation event.

In the context of analyzing orthology data, the problem of reconciling event-labeled gene trees with a species tree appears as a variant of the reconciliation problem where genes trees have no labels in their internal vertices.

When reconciling a gene tree with a species tree, it can be assumed that the species tree is correct or, in the case of a unknown species tree, it can be inferred. Therefore it is crucial to know for a given gene tree whether there even exists a species tree. In this thesis we characterize event-labelled gene trees for which a species tree exists and species trees to which event-labelled gene trees can be mapped.

Reconciliation methods are not always the best options for detecting orthology. A fundamental problem is that, aside from multicellular eukaryotes, evolution does not seem to have conformed to the descent-with-modification model that gives rise to tree-like phylogenies. Examples include many cases of prokaryotes and viruses whose evolution involved horizontal gene transfer. To treat the problem of distinguishing orthology and paralogy within a more general framework, graph-based methods have been proposed to detect and differentiate among evolutionary relationships of genes in those organisms. In this work we introduce a measure of orthology that can be used to test graph-based methods and reconciliation methods that detect orthology. Using these results a new algorithm BOTTOM-UP to determine whether a map from the set of vertices of a tree to a set of events is a symbolic ultrametric or not is devised. Additioanlly, a simulation environment designed to generate large gene families with complex duplication histories on which reconstruction algorithms can be tested and software tools can be benchmarked is presented.

A mis padres
y a toda mi familia

Acknowledgments

The adventure of doing and writing this PhD thesis would not have been possible without the help of an increadible amount of people who probably did not realize the role they played in this process. Thanks to all of you who might not be mentioned here but whom I thank from my heart.

First of all thanks to my advisor Peter F. Stadler for his support of all kinds. You are a great professor but even a greater human. Thanks for your help at all levels, it is something invaluable!

I would like to thank Martin Middendorf for always being there for open discussions and for giving me the oportunity to discover his second home: the beautiful land of Taiwan.

It is of course mandatory for me to thank Nicolas Wieseke, who basically joined me in this adventure all the way, always being available for discussions and to provide new ideas and much of the excitment, thanks a lot Nic for never letting me down!

Somebody who is not less important is Marc Hellmuth, who was open enough to enter and try to understand the world of phylogenomics to be able to get involved in my project. Thanks Marc for the emotions that you injected to the mathematical world of graphs, I learned to love them through you.

Thanks to Sarah Berkemer and Marcus Lechner for putting all their effort and patience to push the last results for my thesis. Sarah, thanks for being willing to change your program as many times as I asked for. Marcus, thank you for running the tools and sending the data as many times as necessary.

Special thanks to all the reviewers, readers and helpers of my thesis: Stephie, Tamas, Mohamed, Ishaan, Steve and all the other anonymous ones. You all did a great job, this thesis looks beautiful because of you all!

The unforgettable are of course Jens and Petra. Petra, thanks for helping me to fill up all the always required forms and for organizing all necessary things for me, including (sometimes) my head. Jens, well, what can I say? my life in this institute would not have been the same without you there, thanks a lot for your help and mainly for your friendship.

This work would have been impossible without the support of all my friends, the ones living in close vecinity and the ones that live far away but are always with me, thanks to all of you for standing by my side. Special thanks to Ric and Cris for providing and discussing ideas, and help with some plots. You are definitely great friends! Thanks to my roommates, to the latino banda, to the beachvolleyball crowd, to the rowing and jogging partners and everybody who make it fun to live in Leipzig.

I would like to thank as well to my former professors Ernesto Bribiesca and Pedro Miramontes for their friendship and for supporting me along my carrer.

Millions of thanks to all the members of my family, who gave up their wish to have me by their side and let me fly without ever lossing the trust in me. ¡Muchas gracias familia! This thesis is dedicated to you all and to the memory of mi abuelita.

Last but not least, thanks to my Stevecito for always being there, for trying to find the right words for me and for his endless support and trust in me. Te quiero muuuucho!

This work was supported by: The *Deutsche Forschungsgemeinschaft* Projekt "Algorithmen zur Rekonstruktion von evolutionären Beziehungen zwischen phylogenetischen Bäumen", The EU-projects EMBIO and EDEN and the Max Planck Institute for Mathematics in the Sciences. Thanks to all of them.

CHAPTER 1

Introduction

Many fields of mathematics have been applied to different areas of biology. In particular, graph theory is the formal basis of the field of phylogenetics. Graphs are a general way of representing biological entities as vertices and the relations between them as edges. For example, two extant organisms may be the descendants of a common ancestor that existed many years ago. Due to adaption processes and genetic drift, the ancestor might have diverged genetically in two different locations or ecological niches, resulting in two new organisms or species. The new species are said to have originated through a *speciation* process. The species can be represented by the vertices of a graph and the relation between parent species and offspring species by the edges of the graph.

Likewise, a gene in an species may occasionally be duplicated in one individual and several generations later this change is found to be a property of all extant individuals of this species. For a single gene, when speciation occurs, we may trace two copies of that gene, one in each new species. Again, we can represent the genes by the vertices of a graph, where two genes are connected by an edge if one is derived from the other, either through *duplication* within a single species or through speciation.

For a larger amount of organisms, related to each other by a number of speciation events, the collection of vertices defines a graph structure known as a rooted tree. This tree is called a *phylogeny* or a *species tree*.

If we consider all the versions of a gene in all the extant species, deriving from a single ancestor gene through a series of duplications and speciations, as vertices and edges in a graph, then we also obtain a rooted tree, which we call a *gene tree*. Gene trees are also a form of phylogeny or *phylogenetic tree*.

Whenever we are able to formalize biological entities like species or genes, in terms of mathematical objects, we are able to leverage mathematical tools to analyse the properties of these objects and

carry over the results to help understanding the underlying evolutionary process. Thus, depending on whether a tree is built with parent and offspring species or ancestral and derived genes, every leaf represents a current organism (a species) or a gene. Based on the hierarchical structure of rooted trees, we can easily see which species (or genes) are more closely related than others. There are several mathematical connections between gene trees and species trees.

A fundamental concept in evolution and genomics is *orthology*, which pertains to the relation between two similar genes, each present in one of two related species. Two genes may have derived from a single ancestral gene form, while the two species have evolved independently from a most recent common ancestor. The two genes are *orthologs* if they are derived from a single ancestral gene that was present in their ancestor species. I.e. the most recent common ancestor of the two related species contained only a single copy of this gene and passed it on to its descendants. If the most recent common ancestor species already had two or more versions of the ancestral gene, then their descendants are not orthologs.

When we want to describe in a phylogenetic tree the evolutionary history of a gene to elucidate its relation with other genes we need a mathematical model to help us to say whether the biological assumptions we made were right or wrong.

Elucidating whether pairs or sets of genes are orthologous to each other is an important task in the reconstruction of evolutionary histories. As yet, however, there has not been a formal mathematical characterization of orthology or criteria for ensuring that orthology relations between sets of genes are being inferred correctly.

In this thesis we present the theory and methods for analysing orthology in order to help scientists develop more accurate algorithms for orthology prediction and have more certainty about their results.

Organization of this thesis

The aim of this work is to define a new mathematical characterization of orthology relations. This implies the study of gene trees, species trees and their reconciliation map. New methods are proposed for the reconstruction of evolutionary gene histories. Simulated data is analyzed to discover a measure of noise in orthology relations based on these new characterizations.

Firstly, basic concepts and definitions that will be used in this thesis are presented in Chapter 2. Then an introduction on the concepts of orthology and paralogy together with the description of the mechanisms involved when those take place are described in Chapter 3. Some existing methods from orthology detection and their differences are also presented here.

We continue in Chapter 4 by presenting the terminology of phylogenetic trees, extending it to the special case of supertrees and rooted triples, as well as some previous results related to these entities that will be of help when presenting the next chapters. The terminology for symbolic ultrametrics as well as the link to phylogenetic trees is also presented.

In chapter 5 we build upon the results of Böcker and Dress [1998] and Semple and Steel [2003] on

symbolic ultrametrics and present new characterizations for them and novel algorithms for recovering the associated trees. An emphasis will be on how these results and algorithms could be potentially used to cope with arbitrary orthology relations. In so doing we shall also show that, somewhat surprisingly, symbolic ultrametrics are very closely related to a well-studied class of graphs called cographs, which is precisely the class of graphs that do not contain induced paths on any subset of four vertices [Corneil et al., 1981].

Continuing with the characterizations, in Chapter 6, we study partitions, cliques and pseudo-cherries that are closely related to the structure of the symbolic representation of orthology relations and present a new algorithm `BOTTOM-UP` to determine whether a map is a symbolic ultrametric or not.

Given that a cograph does not contain the full information on the event-labeled gene tree but it is equivalent to the gene tree's homomorphic image obtained by collapsing adjacent events of the same type one of the results of the previous chapters points out that the orthology relation places strong and easily interpretable constraints on the gene tree.

This observation suggests that a viable approach for reconstructing histories of large gene families may start from an empirically determined orthology relation, which can be directly adjusted to conform to the requirement of being a cograph. The result is then equivalent to an (usually incompletely resolved) event-labeled gene tree, which might be refined or used as a constraint in the inference of a fully resolved gene tree. In Chapter 7 we study the derivation of a species tree from an event-labeled gene tree. As we will show, this problem is much simpler than the full tree reconciliation problem. Technically, we approach this problem by reducing the reconciliation map from gene tree to species tree to rooted triples of genes residing in three distinct species. This is related to an approach that was developed in [Chauve and El-Mabrouk, 2009] for addressing the full tree reconciliation problem.

In Chapter 8 we present a simulation environment designed to generate large gene families with complex duplication histories on which reconstruction algorithms can be tested and software tools can be benchmarked. Using simulations along with the results of previous chapters we introduce a measure of noise in orthology relations, we then test two of the most commonly used graph-based methods `Proteinortho` [Lechner et al., 2011] and `OrthoMCL` [Li et al., 2003] to measure their performance when predicting orthology relations. Using the same measure, we test the database `OMA` to show the accuracy of the data contained in there.

Finally, we present a summary of this thesis in Chapter 9 and propose some ideas for future work and on how all these results can be used to improve orthology prediction methods as well as proposing some ways to modify an "almost" valid orthology relation to a valid one.

Publications

Some results presented in this thesis have been published in the following articles:

- Chapters 5 and 6 introducing new characterizations for valid orthology relations and symbolic ultrametrics are based on the paper:

Marc Hellmuth, Maribel Hernandez-Rosales, Katharina T. Huber, Vincent Moulton, Peter F. Stadler, and Nicolas Wieseke. ***Orthology relations, symbolic ultrametrics, and cographs****. J. Math. Biol., 2013. 66(1-2):399-420.*

- Chapter 7 defining event-labelled genes for which a species tree exists is based on the paper:

  Maribel Hernandez-Rosales, Marc Hellmuth, Nicolas Wieseke, Katharina Huber, Vincent Moulton, and Peter F. Stadler. ***From event-labeled gene trees to species trees****. BMC Bioinformatics, 13(Suppl 19):S6, 2012.*

- The simulation enviroment presented in Chapter 8 was accepted as a poster and won the best poster prize:

  Maribel Hernandez-Rosales, Nicolas Wieseke, Marc Hellmuth, and Peter F. Stadler. ***Simulation of Gene Family Histories****, JOBIM, Paris, France, 2011.*

- Simulations with a modified version of the previous simulations enviroment that integrates synteny information are used from the just submitted paper:

  Marcus Lechner, Maribel Hernandez-Rosales, Daniel Doerr, Nicolas Wieseke, Annelyse Thevenin, Jens Stoye, Sonja J. Prohaska and Peter F. Stadler. ***Orthology Detection Combining Clustering and Synteny for Very Large Data Sets****.* Submitted.

CHAPTER 2

Basic Concepts and Definitions

Graphs, trees, partitions, triples and symbolic ultrametrics play an important role in the analysis of orthology. In this chapter we introduce some notation and concepts of graph theory and ultrametrics.

2.1 Graphs

A *graph* is represented by a set of "objects" where pairs of objects have "connections". In mathematics a graph G is an ordered pair (V,E), where V is a non-empty set of objects called the *vertices* and E is a set of connections called the *edges* and is defined as $E = \{e = \{u,v\} : u,v \in V\}$, where u and v are called the *ends* or *endpoints* of e. If $e = \{u,v\}$ is an edge of a graph, then u and v are *adjacent* or *neighbours*, and e is said to be *incident* to both u and v. A vertex v that exists in a graph but does not belong to an edge is called an *isolated vertex*. The *order* and the *size* of a graph are $|V|$ and $|E|$, respectively. The *degree* of a vertex is the number of edges that connect it to other vertices. An edge of the form $\{v,v\}$ is called a *loop* and is counted twice for the degree of v. A graph that contains multiple edges that connect the same two vertices, is called a *multigraph*. An graph that is not a multigraph and does not contain loops is called *simple*. Unless otherwise stated, we will assume from now on that all graphs in this work are simple.

In an *undirected graph*, the edges have no orientation, since each edge is not an ordered pair but a set $\{u,v\}$ of two vertices. Fig. 2.1(a) shows an example of an undirected graph with vertices $\{a,b,c,d,e\}$.

In a *directed graph*, each edge is represented by an ordered pair of vertices (u,v) and this is called a *directed edge*. In a directed edge (u,v), v is the *head* and u is the *tail* of the edge, since it is considered to be directed from u to v. The number of edges whose tail endpoints are incident to a vertex v is

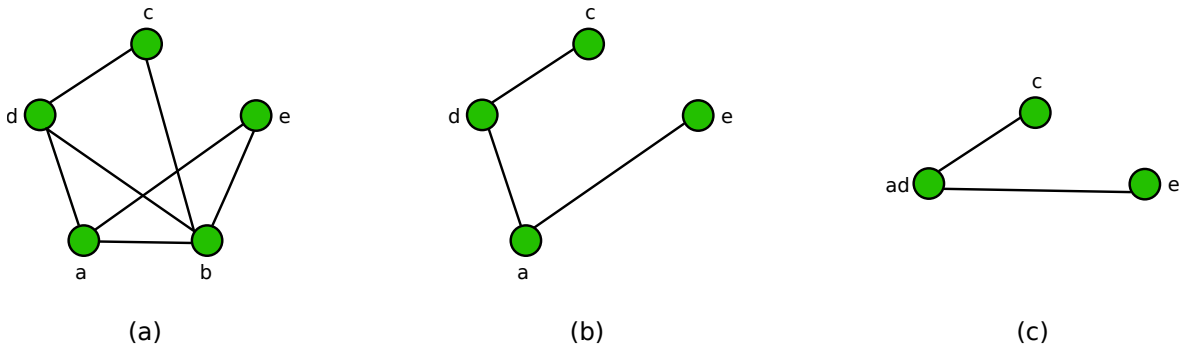


Figure 2.1: **(a)** Undirected graph with vertices represented as green circles and edges as black lines. In this graph an example of neighbors are the vertices $\{a,b\}$. The degree of vertex a is three since there are three edges incident to it. This graph is simple since it does not containt multiedges or loops. **(b)** An induced subgraph on vertices $\{a,c,d,e\}$ from the graph in (a). This graph forms the path c,d,a,e. **(c)** The resulting graph after contracting edge $\{a,d\}$ from the graph in (b).

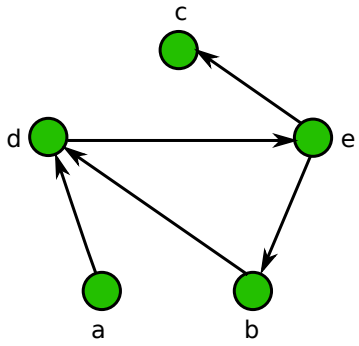


Figure 2.2: A directed graph. Here directed edges are represented by arrows. Edge (d,e) has d as its tail vertex and e as its head vertex. Vertex d has an indegree of two and outdegree of one. If we obtain the induced subgraph on vertices $\{b,d,e\}$, we will obtain the cycle (d,e,b,d).

called the *outdegree* of v and the number of edges whose *head endpoints* are incident to v is called the *indegree* of v. An example of a directed graph is shown in Fig. 2.2.

Two basic operations on edges in graphs are the deletion and contraction. If $e = \{u,v\}$ is an edge of a graph $G = (V,E)$, then $G \setminus e$ is the graph obtained by *deleting* (removing) the edge e from E. The graph G/e is the graph obtained after *contracting* e, that is, by identifying u and v as the ends of e and merging them into a new vertex w, where the edges incident to u and v before contraction will be then incident to w. The new graph will then be $G' = (V',E')$, where $V' = V \setminus \{u,v\} \cup \{w\}$ and $E' = E \setminus e$. Fig. 2.1(c) shows an example of an edge contraction from Fig. 2.1(b). An operation on vertices is the *deletion of a vertex*. $G \setminus v$ denotes the graph obtained by removing v and all the edges incident to it.

A graph $H = (U, D)$ is a subgraph of a graph $G = (V, E)$ if U is a subset of V and D is a subset of E. If a subgraph $A = (W, F)$ of G, is the graph where W is a non-empty subset of V, and F is the set of those edges in E that have both ends in W, then A is the subgraph of G *induced* by W, and is denoted by $A = G[W]$. Fig. 2.1(b) shows an example of an induced subgraph from the graph in Fig. 2.1(a).

A *path* in a graph is a sequence of distinct vertices connected by a sequence of edges. The first vertex of the path is the *start vertex* and the last vertex of the path is the *end vertex*, all other vertices in the path are *internal vertices*, if in a path no vertex is repeated then it is called a *simple path*. A path on n vertices is often referred as P_n. Fig. 2.1(b) shown an example of a simple path. A *cycle* is a path whose start and end vertices are the same.

In a path in a directed graph if the vertex v can be reached from vertex u then v is said to be a sucessor of u and u is said to be a predecessor of v. A *weighted graph* is a graph whose every edge has an associated value called *weight*. The *weight of a path* is the sum of the weights of the edges in the path.

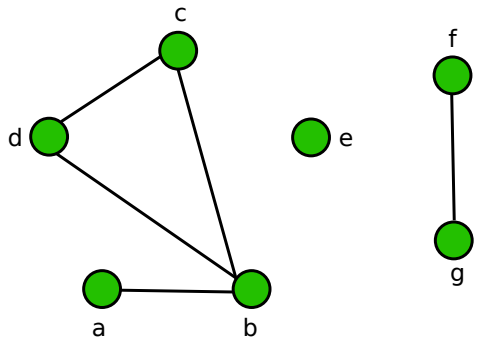


Figure 2.3: A graph with three connected components: $\{a, b, c, d\}$, $\{e\}$ and $\{f, g\}$. The set $\{e\}$ is a singleton.

A graph G is *connected* if for any pair of vertices $\{u, v\}$ there is a path from u to v, otherwise G is *disconnected*. In a disconnected graph, the maximal connected subgraphs are called the *connected components* of G. A connected component consisting of only one vertex is called a *singleton* set. Fig. 2.3 shown a graph with three connected components where one of them is a singleton.

A map f is called *map isomorphism* if for two graphs $G = (V, E)$ and $H = (U, D)$ there is a bijection $f : U \to V$ between the vertex sets U and V, where adjacency is preserved, i.e. $\{u_1, u_2\} \in D$ if $\{f(u_1), f(u_2)\} \in E$, then G and H are said to be *isomorphic*.

In a graph G, if every pair of distinct vertices are adjacent, then G is a *clique*. A clique is also known as a *complete graph*. A clique on n vertices is defined as K_n. Fig. 2.4 illustrates a clique.

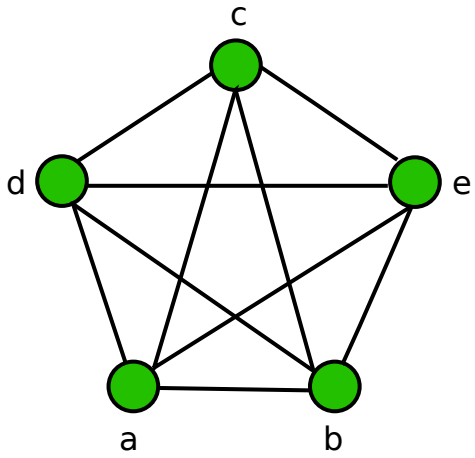


Figure 2.4: A clique on five vertices, a K_5. Every pair of vertices is connected by an edge. A clique is also a cograph since any induced subgraph in four vertices is also a clique and therefore contains no induced $P_4's$.

2.2 Trees

A tree $T = (V,E)$ is a connected cycle-free graph with vertex set $V(T) = V$ and edge set $E(T) = E$. A vertex of T of outdegree zero is called a *leaf* of T and all other vertices of T are called *interior*. A *star* is a tree that has at most one interior vertex. An edge of T is *interior* if both of its end vertices are interior vertices. The sets of interior vertices and interior edges of T are denoted by V^0 and E^0, respectively.

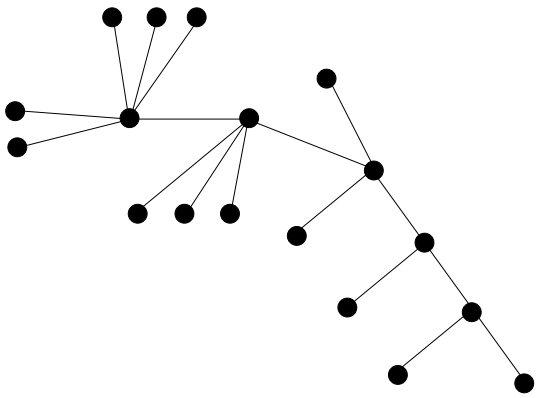


Figure 2.5: An unrooted tree. Leaves have degree one and all other nodes have degree greather than two.

An *unrooted tree* is a tree with no vertices of degree two, like the one shown in Fig. 2.5.

A *rooted tree* $T = (V,E)$ is a tree that contains a distinguished vertex $\rho_T \in V$ with indegree zero, called the *root*. Without explicitly stating it we will always assume that a rooted tree is directed in that all edges of T are directed away from ρ_T. For ease of representation we will always draw rooted trees with the root at the top. A rooted tree T is called *binary* if every interior vertex of T has outdegree two. A *caterpillar tree* is an example of a binary tree that has a central path and deleting all leaves and their incident edges would produce a path. An example of a catepillar tree is shown in Fig 2.6. We define a partial order $\preceq_T$ on V by setting $v \preceq_T w$ for any two vertices $v, w \in V$ for which w is a vertex on the path from v to ρ_T . In particular, if $v \preceq_T w$ we call w an *ancestor* of v and v a descendant of w. A subtree of a tree T is a tree consisting of a vertex v in T and all of the descendants of v in T.

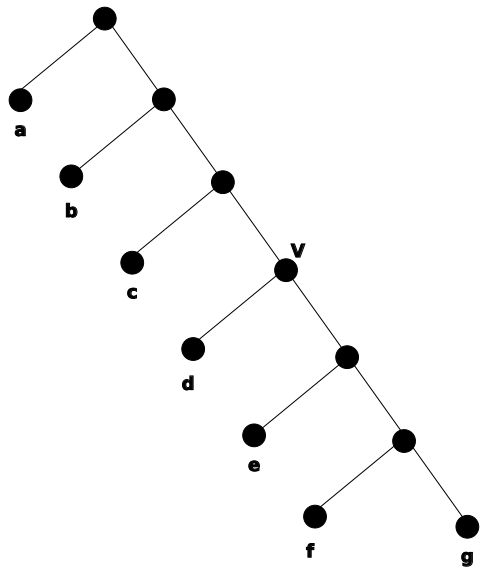


Figure 2.6: A binary tree. This special structure of binary tree is known as a catepillar tree.

A *phylogenetic tree T (on X)* is a rooted tree with leaf set X that does not contain any vertices with in- and outdegree one and whose root ρ_T has indegree zero. The most recent common ancestor *(lca)* of two vertices v and w in T is defined as the lowest vertex u in T such that both $v \preceq_T u$ and $w \preceq_T u$. For $A \subseteq X$ a non-empty subset, we define $\mathrm{lca}_T(A)$, or the *most recent common ancestor of* A, to be the unique vertex in T that is the greatest lower bound of A under the partial order $\preceq_T$. In case $A = \{x,y\}$ we put $\mathrm{lca}_T(x,y) = \mathrm{lca}_T(\{x,y\})$. For $W \subseteq X$ we denote by $T(W)$ the (rooted) subtree of T with root $\mathrm{lca}_T(W)$. For convenience, we will sometimes denote the root of $T(W)$ by ρ_W. Two phylogenetic trees T_1 and T_2 on X are said to be *isomorphic* if there is a bijection $\psi : V(T_1) \to V(T_2)$ that induces a (directed) graph isomorphism from T_1 to T_2 which is the identity on X and maps the root of T_1 to the root of T_2.

In the remainder of this work, X will always denote a finite set of size at least three.

Suppose T is a phylogenetic tree on X with root ρ_T and a non-empty subset $Y \subseteq X$ with $|Y| \geq 2$. Then the *restriction* $T|Y$ of T to Y is the phylogenetic tree obtained from $T(Y)$ by deleting all leaves

$x \notin Y$ and all interior vertices of degree two with the exception of ρ_T if $\rho_T \in V(T(Y))$. For every vertex $v \in V(T)$ we denote by $C(v)$ the subset of X such that $v = \mathrm{lca}_T(C(v))$ and put $\mathscr{C}(T) = \bigcup_{v \in V(T)} \{C(v)\}$. We say that a phylogenetic tree S on X *refines* T, in symbols $T \leq S$, if $\mathscr{C}(T) \subseteq \mathscr{C}(S)$. In addition, we say that T *displays* a phylogenetic tree S on Y if S can be obtained from the restriction $T|Y$ of T to Y by contracting interior edges. $T|Y$ is said to be the *restricted subtree* of T. Fig. 2.7 illustrates this definition. Note that contraction of non-interior edges would not result in a valid phylogenetic tree as such a tree could e. g. have an interior vertex contained in Y.

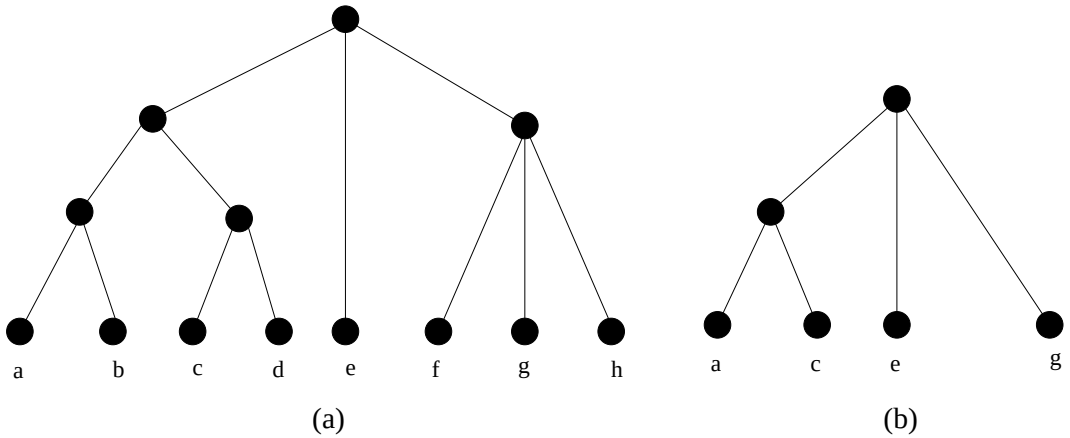


Figure 2.7: **(a)** A phylogenetic tree $T = (V,E)$ on the set $X = \{a,\ldots,h\}$. **(b)** The restricted subtree $T|\{a,c,e,g\}$.

A phylogenetic tree $T = (V,E)$ can have labels on its vertices that represent events, thus we define a map $t : V \to M$, such that M is a set of events and for every $u \in V$, $t(u) \in M$ is the event represented by u. This definition will be useful for the next chapters since a specific case is when an interior vertex of a tree is labeled as "duplication" or "speciation" event. This labeling plays a very important role in orthology analysis.

2.3 Partitions

A *partition* of a set A is the result of dividing A into non-overlapping subsets of A such that for each $a \in A$ is in one and only one of the subsets. These subsets are called *parts* or *blocks*. Equivalently, a set B is called a partition of A if $\emptyset \notin B$, the union of all the blocks in B is equal to A and the intersection of any two blocks in B is empty.

An *equivalence relation* on a set A is a binary relation $\sim$ that satisfies the properties of reflexivity ($a \sim a$), symmetry (if $a \sim b$, then $b \sim a$) and transitivity (if $a \sim b$ and $b \sim c$, then $a \sim c$) for every $a,b,c \in A$. A partition defines an equivalence relation on a set A, when every two elements in a block of the partition are considered to be *equivalent* and each block is defined as an *equivalence class*. In Fig. 2.8 each color represents a block in the partition or an equivalence class.

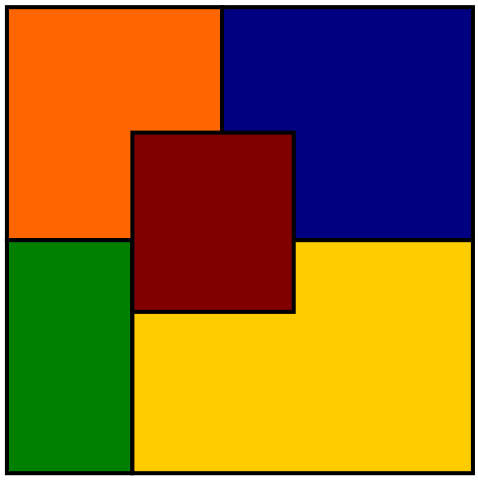


Figure 2.8: A partition of a set into 5 blocks. Each color represents a block and therefore an equivalence class.

2.4 Rooted Triples and Clusters

A *(rooted) triple* is a binary phylogenetic tree on a set Y with $|Y| = 3$. For $Y := \{x,y,z\} \in \binom{X}{3}$, we denote by $xy|z$ the unique triple t on Y with root ρ_t for which $\mathrm{lca}_t(x,y) \neq \rho_t$ holds. Given a phylogenetic tree T on X we denote by

$$\mathscr{R}_T := \left\{ T|Y \,:\, Y \in \binom{X}{3} \text{ and } T|Y \text{ is binary} \right\} \tag{2.1}$$

its set of rooted triples. Note that, for any phylogenetic tree T on X, we have $|\mathscr{R}_T| \leq \binom{|X|}{3}$ and that the maximum is attained precisely if T is binary.

A *cluster* of T is a subset of X whose elements are descendants of a specific vertex of T. In particular, the cluster $\{x,y\}$ is contained in the leaf set $\{x,y,z\}$ of the rooted triple $xy|z$. A *fan triple* is a rooted tree with three leaves and no interior vertices.

2.5 Cographs and Cotrees

Complement-reducible graphs, also named *cographs* [Corneil et al., 1981], are defined recursively as following:

1. K_1 is a cograph.
2. If G is a cograph, then its complement $\overline{G}$ is also a cograph.
3. If G_1 and G_2 are both cographs, then their union $G_1 \cup G_2$ is also a cograph.

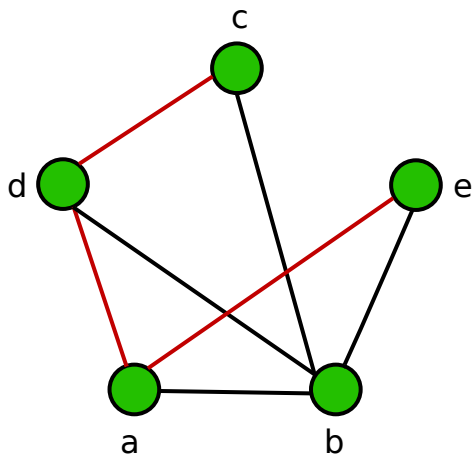


Figure 2.9: A graph that contains an induced P_4, the induced subgraph on vertices $\{a,c,d,e\}$. The P_4 is highlighted in red.

Cographs have been studied extensively, which has led to new characterizations of cographs (see e.g. [Brandstädt et al., 1999] for a survey):

Theorem 1. *[Corneil et al., 1981] Let G be a cograph, then the following statements are equivalent:*

- *G can be constructed from isolated vertices by disjoint union and complementation.*
- *G is a cograph if and only if any induced subgraph on four vertices is not the path P_4.*
- *G is connected if and only if $\overline{G}$ is disconnected.*
- *the complement of any nontrivial connected induced subgraph of G is disconnected.*
- *If G_1 and G_2 are both cographs, then their join $G_1 \bigtriangledown G_2$ is also a cograph.*

In particular, cliques are cographs since every induced subgraph in four vertices will be a clique as well, and therefore it will not contain $P_4's$, like the one shown in Fig. 2.4. An example of a graph that is not a cograph is shown in Fig. 2.9 since the induced subgraph on vertices $\{a,c,d,e\}$ form a P_4, the path $c-d-a-e$.

Furthermore, a graph is a cograph if it can be decomposed in *series* and *parallel* modules. These modules are defined as follows:

Definition 2. *Let $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$ be two graphs, then:*

- *A graph $G=(V,E)$ is a parallel composition of G_1 and G_2 if $V=V_1\cup V_2$ and $E=E_1\cup E_2$.*
- *A graph G is the series composition of G_1 and G_2 if $V=V_1\cup V_2$ and $E=E_1\cup E_2\cup\{\{x_1,x_2\}|x_1\in E_1, and x_2\in E_2\}$.*

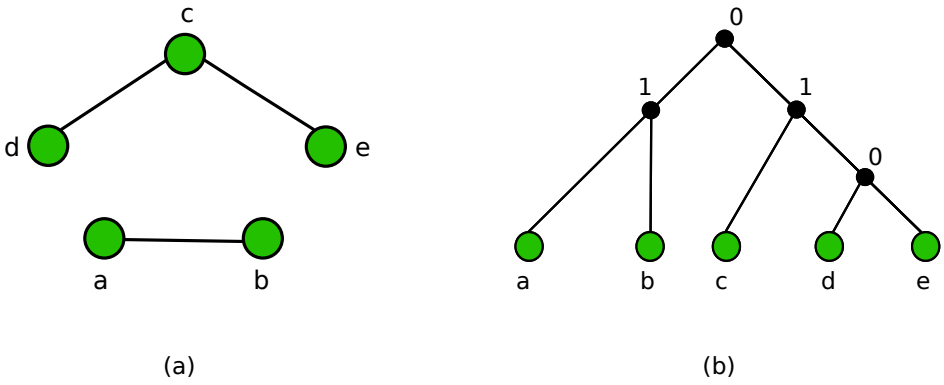


Figure 2.10: **(a)** A cograph. **(b)** The corresponding cotree.

Along with the previous properties, cographs admit a unique tree representation called a *cotree* [Corneil et al., 1981]. A cotree is a tree whose leaves are the vertices of the cograph G and whose internal nodes are labeled either by either 0 or 1. Two vertices in G are connected if and only if their lowest common ancestor in the cotree has a label 1. Moreover, any path from the root to any node of the cotree consists of alternating 0 and 1 labels, Fig. 2.10 shows an example of a cograph and its corresponding cotree: we can see that the path from the root of the cotree to the leaf e consists of alternating 1 and 0 labels. By inverting the labeling of the internal nodes of the cotree, the complement of G is obtained. Nodes with label 1 correspond to series modules, while nodes with label 0 correspond to parallel modules in the modular decomposition.

However, there are other types of modules called *prime modules*, these are modules that are neither parallel nor series. Finding a prime module in a graph means that it contains an induced P_4, implying that cographs do not contain prime modules and that prime modules contain induced P_4's while series and parallel modules do not. These observations are useful for recognizing cographs, which can be done in linear time as well as the computing of the corresponding cotree [Corneil et al., 1985].

2.6 Ultrametrics and Tree Metrics

A *metric* is a distance function that defines a distance between elements of a set. The distance function on a set X is defined as:

$$d : X \times X \to \mathbb{R}$$

where $\mathbb{R}$ is the set of real numbers.

For all $x, y, z \in X$, d must satisfy the following conditions:

1. $d(x,y) \geq 0$
2. $d(x,y) = 0$ if and only if $x = y$

3. $d(x,y) = d(y,x)$

4. $d(x,z) \leq d(x,y) + d(y,z)$

Condition 4 is called the *triangle inequality*.

An *ultrametric* is a metric that satisfies the following stronger version of the triangle inequality. For all $x,y,z \in X$,

$$d(x,z) \leq max\{d(x,y), d(y,z)\}.$$

An ultrametric on X can be seen as a matrix whose rows and columns correspond to the elements of X. Thus, a matrix d_{ij} is an ultrametric on X if and only if all the elements d_{ii} in the diagonal are zero, and it satisfies:

$$d_{ij} \leq max\{d_{ik}, d_{kj}\}.$$

for all $i,j,k \in X$.

A *tree metric* is a matrix d_{ij} with zero diagonal which satisfies the *four-point condition*:

$$d_{ij} + d_{kl} \leq max\{d_{ik} + d_{jl}, d_{il} + d_{jk}\}$$

for all $i,j,k,l \in X$.

By checking the possible configurations of paths which can connect four points x,y,z,w in a tree, it can be seen that the distance function satisfies the inequality:

$$d(x,y) + d(z,w) \leq max\{d(x,z) + d(y,w), d(x,w) + d(y,z)\}.$$

Every ultrametric is, therefore, a tree metric, and a tree metric can be characterized in terms of an associated ultrametric [Bandelt, 1990].

2.7 Symbolic Ultrametrics

Let $T = (V,E)$ be a compact rooted tree with leaf set X. T is said to be *dated* by a map $t : V \to \mathbb{R}$ if $t(x) = 0$ for all $x \in X$, and $t(v) \prec t(u)$ for every edge $(u,v) \in E$.

Let M denote an arbitrary non-empty finite set. A map $t : V \to M$ is called a *symbolic dating map*.

Now, let $\odot$ be a special element not contained in M, and $M^{\odot} := M \cup \{\odot\}$. The symbol $\odot$ corresponds to a "non-event" and is introduced for purely technical reasons. It will always correspond only to the leaves of T since these will not usually correspond to events such as speciation and duplication.

Now, suppose $\delta : X \times X \to M^{\odot}$ is a map. We call δ a *symbolic ultrametric*[1] if it satisfies the following conditions:

(U0) $\delta(x,y) = \odot$ if and only if $x = y$;

[1] Note that in [Böcker and Dress, 1998] a symbolic ultrametric is defined without the requirement (U0), which we have introduced for technical reasons.

(U1) $\delta(x,y) = \delta(y,x)$ for all $x,y \in X$, i.e. δ is symmetric;

(U2) $|\{\delta(x,y), \delta(x,z), \delta(y,z)\}| \leq 2$ for all $x,y,z \in X$; and

(U3) there exists no subset $\{x,y,u,v\} \in \binom{X}{4}$ such that

$$\delta(x,y) = \delta(y,u) = \delta(u,v) \neq \delta(y,v) = \delta(x,v) = \delta(x,u). \tag{2.2}$$

Note that every symmetric map δ on X with $|X| = 3$ that also satisfies Properties (U0) and (U2) is as well a symbolic ultrametric on X.

CHAPTER 3

Homologs, Paralogs and Orthologs

The variation coded in the genomes of a group of living species, which may have evolved for many millions of years, contains a wealth of information on how these species have diverged from their common ancestor. Before whole genome sequences became available, this information pertained mostly to the evolution of one gene at a time, relating to one protein or RNA molecule as it accumulated mutations in various evolutionary lineages.

The realization that many genes reoccurred in two or more highly similar versions in a single genome, due to various duplication processes, prompted the study of *gene families*, although lacking complete genomes, uncertainty about the total membership of a gene family was a major hindrance.

3.1 Homology

The availability of many complete or nearly complete genome sequences, due to rapid new sequencing technology, allows us access to much higher quality data that were previously available in only very rudimentary form. Two important aspects of genome-scale data are

- A near-complete inventory of all the genes or putative genes in the genome, as well as other structural elements, ordered along the chromosomes.
- A partition of the genes in one or more genomes into gene families, as defined by an analyst's choice of threshold of gene similarity, i.e, a criterion of *homology*.

Homologous regions of genomic sequence in one or more genomes are regions which are believed to have had a relatively recent common ancestry. More specifically, *homologous genes* are inferred to have descended from a common ancestor.

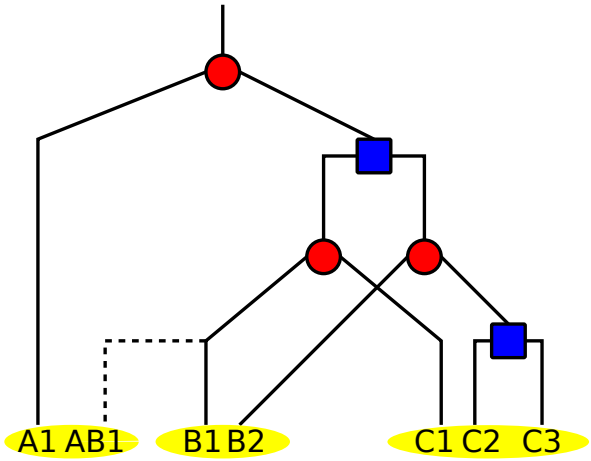


Figure 3.1: The evolution of a gene. Extant species A, B and C (in yellow elipses) contain instances of genes after duplications and speciations. Speciations are depicted as red circles and duplications as blue squares. Horizontal gene transfer is depicted as a dashed line from species B to species A. (Figure adapted from Fitch 2000 [Fitch, 2000])

We can distinguish at least three kinds of homology:

- *Orthology* pertains to two homologous genes whose divergence stems from a *speciation* event. Orthologs have the property that the phylogeny of a set of genes is the same as the true phylogeny of the organisms where they reside [Fitch, 1970].

- *Paralogy* relates genes whose ancestors diverged from each other starting at a *duplication* event within a single ancestral genome [Fitch, 1970].

- *Xenology* is the result of horizontal transfer, where one of two homologous genes is the result of an interspecies transfer of genetic material [Gray and Fitch, 1983].

Homology should be distinguished from *analogy*, a similarity between two genes that have descended from *different* ancestors but are similar at the sequence level and/or perform the same function, as result of convergent evolution.

Fig. 3.1 depicts a gene tree illustrating the three kinds of homology, orthology, paralogy and xenology. Gene $A1$ is orthologous to all genes in species C. All genes $C1$, $C2$ and $C3$ are paralogous to each other, however only gene $C1$ is orthologous to gene $B1$ in species B, similarly, gene $B2$ is orthologous to genes $C2$ and $C3$ but not to $C1$. Xenology is created by the horizontal gene transfer from species B to species A.

In this chapter and throughout this thesis, we will focus on paralogy and orthology.

3.2 Orthology and Paralogy

The distinction between paralogous and orthologous genes often correlates at the level of gene function. After a speciation event, the two orthologs descending from a single gene in the ancestor must continue to fulfill the same function. After a duplication event, however, the two paralogs thus created in a single genome need not both conserve exactly the same function. As long as one copy continues to perform the original function, the other may diverge to carry out a different function, or may lose functionality completely. Alternatively the two paralogs may share, or divide up between themselves, the original function.

Orthologous genes belong to different species, by definition. Paralogous genes are usually thought of as belonging to the same species, but this is not always the case. If a speciation event occurs after a duplication, then genes from the two daughter species may be paralogous as well. In Fig. 3.1 one can observe than even if genes B1 and C3 belong to different species, they are paralogs since they are the result of an earlier duplication.

Additional terminology pertaining to homology relations refer to the temporal sequencing of two events [Kristensen et al., 2011]:

- In-paralogy: paralogs that arose by duplication after a specific speciation event.
- Out-paralogy: paralogs that arose by duplication before a specific speciation event.
- Co-orthology: in-paralogs that collectively are orthologous to genes in other organisms.

3.2.1 Functional divergence

In this section we analyse the evolutionary mechanisms of diversification of paralogous genes.

When gene duplication occurs, changes in each copy generally occur independently, although there may be repair and conversion mechanisms that tend to keep them similar as well as selection at the functional level that ensures the viability of the organism.

The changes are normally point mutations or indels in the gene that could result in functional novelty or in the loss of function of one of the paralogs.

Three types of mechanisms [Dittmar and Liberles, 2010] operating on duplicated genes, illustrated in Fig. 3.2, are:

- Pseudogenization
- Subfunctionalization
- Neofunctionalization

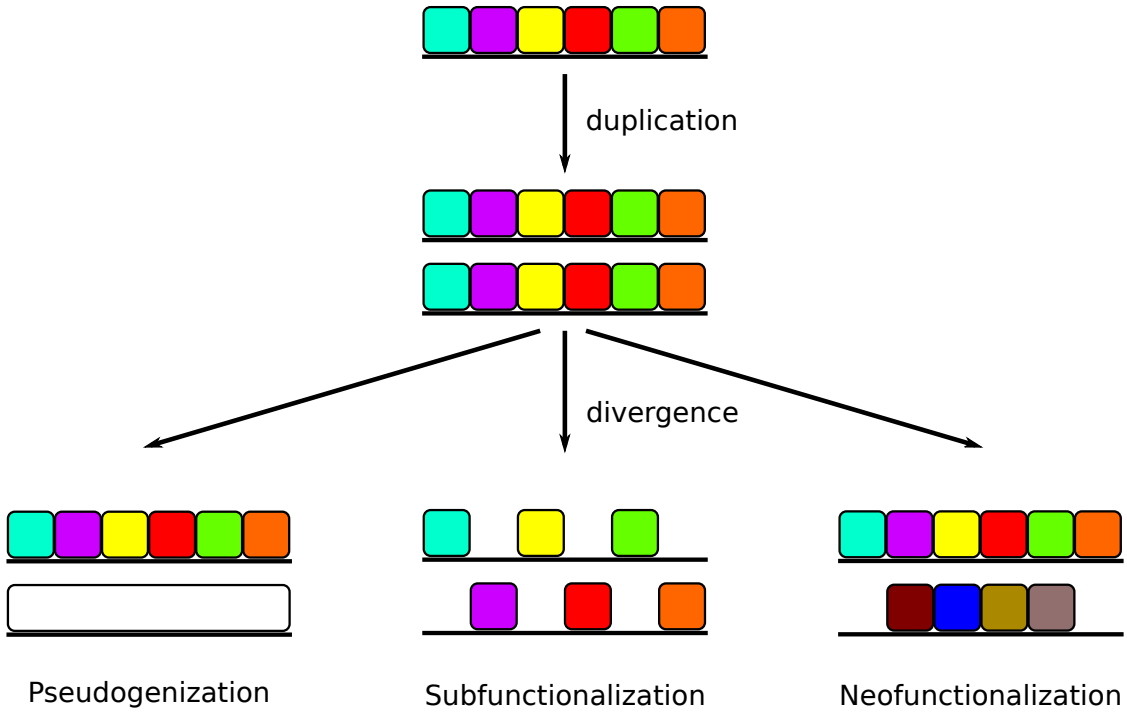


Figure 3.2: Functional divergence. Pseudogenes are genes that are not functional and not necessary for the survival of the organism where they reside. Subfunctionalization gives rise to division of labor of the new paralogs, each new copy will take a different subfunction of the original ancestral one. Neofunctionalization occurs when one of the paralogs takes a complete new function that the ancestral gene did not have and the other paralog retains the original ancestral function.

Pseudogenization

Pseudogenization, sometimes referred as gene degeneration or gene loss, since it gives rise to genes (pseudogenes) that are not functional, typically due to mutations producing stop codons within previously coding regions. This type of gene is not protein-coding and its product is not necessary for the survival of the organism, since it continues to be produced by the other copy of the gene.

Subfunctionalization

Some genes are wholly responsible for a general function throughout the organism at all developmental stages. After a gene duplication, the new paralogs can undergo division of labor and take on different subfunctions of their original ancestral function. Each new copy will only retain a specialized version of the original function, say at a particular developmental stage or in a specific tissue. This mechanism is called subfunctionalization. Genes affected by several regulatory regions are more prone to subfunctionalization. An example is shown in Fig. 3.2, here each function is depicted with a different color. The new paralogs take some of the subfunctions of the original ancestral gene.

Neofunctionalization

Neofunctionalization is associated with the formation of new functions. After gene duplication, neofunctionalization can occur if one of the paralogs takes a complete new function. This gene must have been mutated to develop a function that the ancestral gene did not have while the other paralog retains the original ancestral function.

3.3 Small and Large Scale Gene Duplications

Small scale duplications refer to processes that result in the duplication of one or few genes, or sometimes even just a part of a gene. When genes are duplicated, they may be fixed in the population to become an invariant part of the genome.

Occasionally large scale duplication occurs, whereby a whole genome, one or more whole chromosomes, or a large chromosomal segment is duplicated. Typically this type of duplication is followed by massive gene loss and a period of intensified genome rearrangement.

Because of mutational divergence and functional change it is often difficult to distinguish paralogous genes from orthologous genes. Several methods have been developed over the years. In the next section we will discuss some of the most important of these.

3.4 The Orthology–Paralogy Distinction

Sets of orthologous genes are used to explore evolutionary history and to infer phylogenies, largely on the basis of nucleic acid and protein sequence divergence. Using paralogs, mistakenly identified as orthologs, in two species instead of true orthologs leads to errors in phylogenetic inference, especially with respect to speciation events for which these paralogs are out-paralogs.

Tree-based methods for distinguishing between orthologs and paralogs are designed to avoid such problems. These methods often rely on the comparison of a gene tree with a species tree.

If the gene trees and species trees have been constructed accurately, the identification of orthologs is straightforward, namely the pairs of genes that diverge as result of speciation of their most recent common ancestor.

In the following section we present some of the tree-based methods available for this purpose.

3.4.1 Tree-based Methods

Methods based on gene tree/species tree reconciliation involve the identification of every internal node of the gene tree as a duplication or speciation event, by taking into account the phylogeny of the species tree. The gene tree with each internal node labeled as duplication or speciation is called the *reconciled tree*. From this tree it is straightforward to deduce orthologs and paralogs, so that the distinction between orthology and paralogy can be reduced to tree reconciliation. Fig 3.3 shows an example of a species tree, a gene tree and their reconciliation tree.

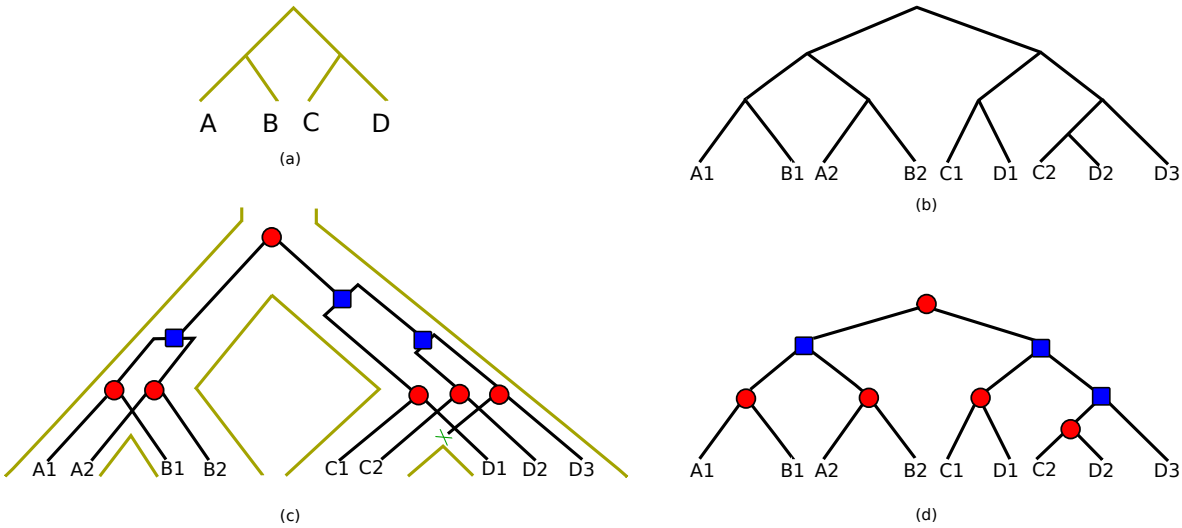


Figure 3.3: **(a)** A species tree.**(b)** A gene tree.**(c)** The reconcilied tree, the gene tree is embeeded in the species tree. **(d)** The reconciled tree with duplication/speciation events at the internal nodes. Red circles represent speciation event, blue square duplications, a gene loss is represented with a green cross.

Most reconciliation tree methods rely on parsimony: the reconciliation that requires the least number or duplications and losses is suggested as the solution to the reconciliation problem. Some methods may weight duplications and losses differently.

The reconciliation problem has some limitations in practice which have given rise to refinements to solve them. In Table 3.1 we summarize some of them and in the following sections we discuss them in more detail.

Unrooted trees

Reconciliation methods often require that the species tree and the gene tree be rooted. Frequently, however, rooting information is not directly available. To root the gene tree Hallett and Lagergren [2000] adopted the principle of parsimony. The root is chosen in such a way that the gene tree has the minimum number of duplications. Storm and Sonnhammer [2002] present `Orthostrapper` a method that analyzes a set of bootstrap trees to estimate orthology support values from pairs of sequence in a multiple aligment. If the gene tree is unrooted, they place the root at the center of tree based on the idea that there is a molecular clock [Farris, 1972]. A similar method is implemented in the software package `RIO` [Zmasek and Eddy, 2002], which estimates how reliable orthology assignments are by performing analyses over bootstrap-resampled phylogenetic trees. For the case of multiple optimal rootings the method will select the tree that minimizes the tree height.

In the case of unrooted species trees, the rooting approach most often used is the identification of an outgroup species. A reliable outgroup must be closely enough related to share significant DNA

Method	Description
Orthostrapper	Uses bootstrap trees to calculate orthology. If the gene tree is unrooted, a midpoint is calculated and the root is placed here.
RIO	Estimates the reliability of orthology assignments by performing analyses over bootstrap-resampled phylogenetic trees. In the case of multiple optimal rootings the method selects the tree that minimizes the tree height.
TreeBeST	Integrates multiple tree topologies, combining this with a species-tree aware penalization of those topologies inconsistent with known species relationships. Treats the problem of species uncertainty by treating ambiguous regions of the tree as multifurcarting nodes.
LOFT	A species trees is not needed. The method makes use of the species-overlap method, constructing hierarchical groups that highlight relatedness differences between orthologous and paralagous genes.
PhylomeDB	Alignment trimming phases and evolutionary models are implemented in a pipeline which makes use of the species-overlap method.
MetaPhOrs	Integrates information from multiple phylogenetic trees obtained from different sources. Makes use of the species-overlap algorithm.
COCO-CL	Explores the correlation of evolutionary histories of homologous genes in a more global context by using a measure of sequence distance.
HOGENOM	For a given tree topology, infers speciation and duplication events and then identifies probable orthologs and paralogs. If the gene tree has unresolved regions, it collapses them into multifurcating nodes.
Softparsmap	Algorithm that enables soft parsimony by the mapping of gene trees onto species trees and subsequent modification of uncertain or weakly supported branches. The method takes account of low bootstrap values while minimizing the number of gene duplication and loss events.
TreeFam	Orthologs and paralogs are inferred from the phylogenetic tree of a gene family. After the automatic generation of gene trees, these are curated by experts.
PHOG	Relies on pre-computed gene trees in the first step and thereafter on tree distance thresholds that can be defined by the user.

Table 3.1: Tree-based methods for orthology inference.

or protein homology with the species under study, but it must be clear that it is less closely related to these than they all are to each other. Huerta-Cepas et al. [2007] make use of this method for the rooting of gene trees.

Uncertainty in species trees

Most reconciliation methods rely on the correctness of the species tree. Often, however, these trees contain uncertainties. TreeBeST [Li et al., 2006] tackles this problem by treating ambiguous regions of the tree as multifurcating nodes. This is integrated into the EnsemblCompara project [Vilella et al., 2009].

PhylomeDB [Huerta-Cepas et al., 2007] and MetaPhOrs [Pryszcz et al., 2011] adopted an approach

that does not require a species tree. This approach identifies, for each given internal node in the gene tree, the set of species represented in each subtree of the node. If the intersection of the sets is empty, then a speciation event is inferred, otherwise a duplication event is. This method, which was called *species overlap* by the authors was first implemented in LOFT [van der Heijden et al., 2007]. A similar method for unknown species trees is implemented in BranchClust [Poptsova and Gogarten, 2007]. This relies on the identification of subtrees in gene trees. These subtrees consist of sequences found in most species and the method in effect delineates COGs-like clusters.

A different approach that also constructs clusters, but in a hierarchical way, is implemented in COCO-CL [Jothi et al., 2006]. This method explores the correlation of evolutionary histories of homologous genes in a more global context by using a measure of sequence distance.

Uncertainty in gene trees

Another problem in reconciliation methods is the assumption that the gene tree is correct. As with uncertainty in species trees, however, this assumption does not always hold true. In HOGENOM Dufayard et al. [2005] proposed to take care of this problem by collapsing unresolved parts of the gene tree into multifurcating nodes, in the same way as it is done for species trees. In Softparsmap Berglund-Sonnhammer et al. [2006] propose a similar approach by collapsing only branches with low bootstrap values.

A more elaborate approach is adopted by TreeFam [Li et al., 2006]: after the automatic generation of gene trees, these are curated by experts. As orthologs and paralogs are inferred from the phylogenetic tree of a gene family and not from BLAST matches, curators only edit a tree if additional considerations, such as in gene function analysis, strongly suggest that the automatically generated tree is incorrect. PHOG [Datta et al., 2009] is another method that relies on pre-computed gene trees in the first step and then on user-defined tree distance thresholds. For each sequence (leaf in the gene tree), the closest sequence in each other species is identified by a tree distance defined by the sum over the edge lengths. In this way, this method obviates the need for a species tree, since it only requires the list of species, the tree topology and the tree distances.

3.4.2 Graph-based Methods

Explicit phylogenomic analysis may be the best approach to the inference and differentiation of orthologs and paralogs. Trees, however, have the disadvantage that constructing them is computationally expensive when the number of leaves is large. Moreover tree reconstruction is sensitive to noise, so that when automated methods are used for large number of sequences at large evolutionary distances the output can be biased by the inaccuracy of multiple sequence alignments [Felsenstein, 2004]. A fundamental problem is that, aside from multicellular eukaryotes, evolution does not seem to have conformed to the descent-with-modification model that gives rise to tree-like phylogenies. The evolution of prokaryotes and viruses seems to have involved a substantial component of horizontal gene

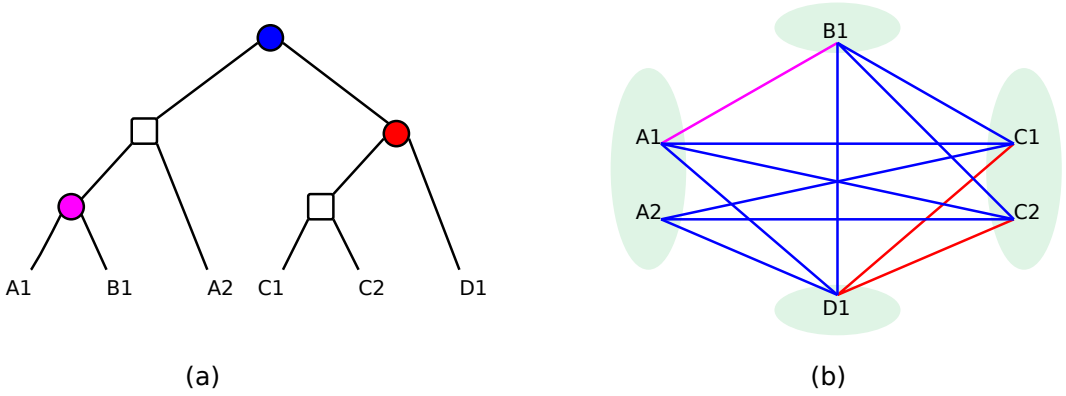


Figure 3.4: **(a)**. An evolutionary scenario with three speciation events represented by circles and two duplication events represented by squares. **(b)**. The orthology graph. Each oval represents a species. The color of the edges between genes represents the corresponding speciation in the tree in (a) that makes the pair of genes to be orthologs. Due to a duplication after speciation, in-paralogs $C1$ and $C2$ are both orthologs to gene $D1$. Here one can observe $1-to-1$ orthology relationship between genes $A1$ and $B1$, $1-to-many$ between gene $D1$ and genes $C1$ and $C2$ and $many-to-many$ between genes $A1$, $A2$ and genes $C1$, $C2$.

transfer [Puigbo et al., 2010]. A graph representation of their evolutionary history, such as a reticulated tree or even a more general structure, would then be more appropriate in this case. To treat the problem of distinguishing orthology and paralogy within this more general framework, graph-based methods have been proposed to detect and differentiate among evolutionary relationships between genes in those organisms. Nodes in the graphs represent genes, and edges the paralogy or orthology relationship. Fig. 3.4 shows an example of this approach.

Methods motivated by this approach typically run in two phases:

- The graph construction phase: pairs of orthologs are inferred.
- The clustering phase: clusters of orthologs are constructed.

Approaches for the graph construction phase

Using sequence similarity scores as a measure of relatedness, algorithms identify orthologous genes for each pair of genomes. This is commonly done by identifying, for each gene, its ortholog in the other genome using the criterion of bidirectional best hit (`BBH`). This requires, for a candidate pair of orthologs a, b, that a is the best hit for b and vice versa, that b is the best hit for a. In Fig 3.5 a scenario of the `BBH` approach is pictured. Due to the limitations of the `BBH` approach, refinements have been developed, as found in Table 3.2.

In `Inparanoid` [Remm et al., 2001] the authors improve on the `BBH` approach by identifying one-to-many and many-to-many orthologous relationships. If a duplication event takes place after the

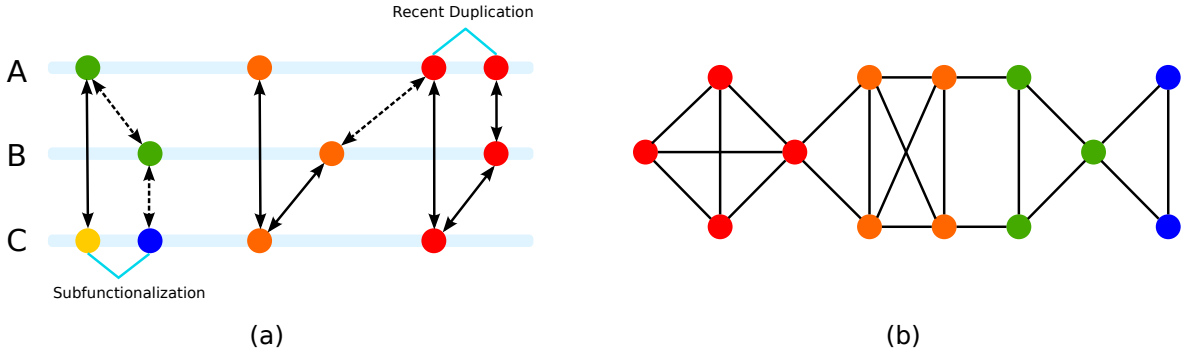


Figure 3.5: An escenario using the BBH approach to identify orthologous genes. (a) Three horizontal bars represent three different species. Circles on each bar represent genes belonging to that species. Colors of the circles indicate a certain biological function; same colors indicate the same biological function. Black bi-directional arrows represent BBHs: a solid BBH arrow means a true positive, i.e., it links two genes with the same function, and a dashed BBH arrow means a false positive, i.e., it links two genes with different functions. Duplicated genes are connected by blue lines. Genes are arranged into three columns on the panel. The first column includes four genes. The two green genes from species A and B is a pair of true positive BBH. There is a duplication event that caused a subfunctionalization event in species C, i.e., the original green function is shared by the blue and yellow functions in this species. Green gene from species A is connected through a BBH linkage to the yellow gene in species C, but their function are not identical. Similarly, green gene in species B is connected to blue gene in species C. Here, subfunctionalization results in two false positive BBH linkages. In second column, there are three orange circles, which should have been all connected by true positive BBHs. However, if the function corresponding to the orange circle has some relationships with that corresponding to red circle at the third column, the orange gene from species B and a red gene from species A are detected as a pair of BBH. This is an example of false positive, which is shown as a dashed BBH arrow. The third column is is a group of four red circles representing four genes with identical functions. There is a recent gene duplication event in species A, which creates two paralogs (two red circles on the first bar) with the same biological function. (b) A network showing the topology of a plausible ortholog group. Nodes are genes and edges are BBH linkages. There are four different functions in this ortholog group (indicated by the four colors). Further partition work is required. (This figure is a partial self-reproduction of one in [Fang et al., 2010]).

speciation event at the most recent origin of both genomes under study, the result will be a set of in-paralogs, all as orthologs to one or more genes in the other organism, depending whether duplication occurs in one or both lineages. OrthoInspector [Linard et al., 2011] is a software system that also incorporates in-paralog detection before investigating the BBH between pairs of groups. In addition, it detects contradictory information between groups.

Another approach that makes use of the BBH is implemented in the pipeline DODO (Domain-based detection of orthologs) [Chen et al., 2010]. In a first step, DODO classifies proteins into groups based

Method	Description
Inparanoid	Identifies BBH for pairs of organisms and then applies statistical rules to identify in-paralogs that might be merged due to duplication after speciation.
OrthoInspector	software system that detects in-paralogous groups and then investigates the 1-to-1, 1-to-many and many-to-many BBH between pairs of groups. In additional, it detects contradictory information between groups.
DODO	Uses domain-based detection of orthologs to classify proteins into groups and then applies BBH to each of them.
Roundup	Calculates maximum likelihood of the evolutionary distance between pairs of sequences.
EggNOG	Contains orthologous groups constructed hierarchically through identification of BBH and triangular linkage clustering.
OrthoDB	Hierarchical catalog of orthologs.
OrthoMCL	Uses a Markov clustering involving iterative simulations of randomized flow on edges of the orthology graph obtained by BBH.
ProteinOrtho	Applies an adaptive best match method together with spectral clustering to define co-orthologs.

Table 3.2: Graph-based methods for orthology inference.

on protein domain information. This approach is motivated by the idea that domain composition is more likely conserved through evolution than similarity between primary orthologous sequences. Thereafter, orthology relationships are refined by identifying the BBH in each protein group.

An alternative to sequence similarity measures is the calculation of maximum likelihood of the evolutionary distance between pairs of sequences. This approach is implemented in Roundup [Wall et al., 2003] and is motivated by studies that have shown that often the nearest phylogenetic neighbor is not the one that obtains the closest BLAST hit in the alignment [Koski and Golding, 2001].

Approaches for the ortholog clustering phase

The concept of Clusters of Orthologous Groups (COGs) was first introduced by Tatusov et al. [1997]; the motivation was to work with clusters of orthologs instead of lists of orthologous pairs. The clustering algorithm first identifies all triplets of genes connected to each other (triangles) and then merges them if they share a common face. When there are no more triangles to be added, the algorithm stops. The idea behind this approach is to cluster genes that have diverged from a single gene belonging to the most recent common ancestor of the species involved.

OMA [Altenhoff et al., 2011] is a database where, for two genomes, clusters of genes are defined as groups where all pairs of genes are orthologs. To avoid errors when classifying orthologs, and not to confuse them with paralogs, when an ortholog is missing the method verifies what the authors define as “stable pairs” [Roth et al., 2008] with sequences in a third genome that can act as “witnesses” of evolution.

Hierarchical clustering is an approach adapted by EggNOG [Jensen et al., 2008] and OrthoDB [Wa-

terhouse et al., 2013], where groups of orthologs are clustered with respect to a particular speciation. In this way, the groups will contain orthologs and in-paralogs with respect to that particular speciation. Clustering is performed by triangular linkage in both methods.

A different clustering approach is presented by `OrthoMCL` [Li et al., 2003], making use of Markov Clustering [van Dongen, 2000]. The process starts by simulating a random walk in the weighted graph where each edge has an orthology score. Later the Markov Clustering calculates the probability of two genes to belong to the same cluster. According to these probabilities the graphs is then partitioned into orthologous groups.

A program that can be run in stand-alone mode is `ProteinOrtho` [Lechner et al., 2011] whose clustering method focuses on analysing an edge-weighted directed graph. The result is disjoint complete multipartite subgraphs where a species is represented at most once in each of them, therefore each subgraph represents a set of orthologs.

3.4.3 Synteny

Most of the methods mentioned in the previous sections rely on sequence similarity and do not take into account local gene order (synteny) that might provide valuable evolutionary information. Synteny is more conserved between closely-related organisms. When a set of orthologs surround a set of homologous genes, it is likely that those homologs are orthologs as well [Jun et al., 2009].

Synteny information has been introduced as a second step in several methods to increase confidence in orthology prediction. After investigating each gene's neighbourhood only genes in similar neighbourhoods are kept as potential orthologs.

`OrthoParaMap` [Cannon and Young, 2003] and `PhyOP` [Goodstadt and Ponting, 2006] are orthology prediction tree-based methods that combine phylogenetic trees with synteny information between pairs of closely related species.

`SYNERGY` [Wapinski et al., 2007] infers orthology of all genes among a large group of species and uses synteny information when available for all the species. The identification of mammalian orthologs using local synteny was performed by `MSOAR` [Shi et al., 2010]. This approach incorporates tandem duplication information based on genome rearrangement when assigning orthology.

The *Encapsulated Gene-by-gene Matching* (`EGM`) [Mahmood et al., 2010] is a graph-based method that identifies orthologs and conserved gene segments for pairs of genomes. `EGM` constructs a global gene matching with maximum weight in the bipartite graph taking into account gene context, orientation information and sequence similarity. This matching will result in a one-to-one correspondence between putative orthologs in the pair of genomes.

A similar approach is proposed by Sankoff [Sankoff, 2011], where pairwise synteny blocks are constructed as a first step, making use of sequence similarity. This is followed by the combination of all the possible orthologs thus identified from all pairs of species into sets from which optimal multipartite subgraphs may be extracted. Here a species is represented at most once in each subgraph, so that it represents a set of orthologs.

3.5 Concluding Remarks

Identification of orthologous genes is important in elucidating the evolutionary history of species and their genes. Here we have presented several methods for orthology identification that fall into two main classes: tree-based methods and graph-based methods. In both classes synteny information can be added to increase confidence in orthology prediction. Tree-based models employ specific models of evolution and therefore the identification of orthologs, co-orthologs, paralogs and in-paralogs falls out naturally. The disadvantages of these methods are that they are computationally demanding and that not all organisms have a tree-like evolutionary history, in particular the prokaryotes and viruses. In such cases graph-based methods are more suitable.

We have seen that the nature of duplication mechanisms lead to several difficulties in distinguishing between paralogs and orthologs. In the course of this work we will focus on evolutionary histories represented by trees. Once a reconciled tree is reconstructed the identification of orthologous and paralogous genes is straightforward; but we can never be sure that the reconstructed reconciliation tree is the right one. Moreover, for a given set of genes and species one would like to know beforehand whether it is even possible to reconstruct their evolutionary history.

In this thesis we study the mathematical properties of species trees, gene trees and their reconciliation trees to characterize valid orthology relations and a valid mapping of the gene tree onto a species tree.

CHAPTER 4

Mathematics of Phylogenies

The reconstruction of the evolutionary history of a set of genes and a set of species is still a challenge in phylogenetics. Therefore tree reconstruction is an important task. In this chapter, we present some previous results on phylogenetic trees and rooted triples which play an important role determining properties of phylogenetic trees and supertrees. Here we also present a summary of supertree methods for the reconstruction of large phylogenetic trees. Surprisingly, phylogenetic trees have a 1-to-1 correspondance to symbolic ultrametrics and some properties of this relation are presented as well.

4.1 Introduction

A phylogenetic tree is a tree that represents evolutionary relationship between species. Internal nodes represent speciation events and the branching structure of the tree reflects the species tree evolution. All the species in the phylogenetic tree have one common ancestor, this is depicted as the root of the tree. Extant species are depicted as the leaves of the tree.

Phylogenetic trees are used to represent tree-like evolutionary histories. The inference of a phylogenetic tree that contains a large number of leaves is a difficult task due to the computational complexity that this requieres.

An approach that has been used more often in the recent years is the divide-and-conquer-based *supertree method* [Bininda-Emonds, 2004a, Rauch Henzinger et al., 1999, Jiang et al., 2002]. A supertree is the result of merging many smaller overlapping phylogenetic trees into a single larger tree.

In the following sections, we present some mathematical results of phylogenetic trees and su-

pertrees. Unless stated otherwise, we will follow the notation in [Semple and Steel, 2003].

4.2 Supertrees

Supertrees have been used to combine rooted phylogenetic trees with overlapping leaf sets. The most widely used method is Matrix Representation with Parsimony analysis (*MRP*) [Ragan, 1992, Baum, 1992]. This supertree method analyzes multiple trees whose internal nodes are ordered hierarchically. This method is based on nodes instead of on full trees. This allows to combine data sets with not necessarily the same taxa. However the taxa can be overlapping [Baum, 1992]. Other more recent methods have been developed, examples are: *MRD* (Matrix Representation with Distances) [Lapointe and Cucumel, 1997], *MRC* (Matrix Representation with Flipping) [Eulenstein et al., 2004], *MRC* (Matrix Representation with Compatibility) [Purvis, 1995], and *Bayesian Supertrees* [Bininda-Emonds, 2004b].

Aho et al. [1981a] have given a polynomial algorithm for determining whether a set of rooted phylogenetic trees is compatible. In [Semple and Steel, 2003] this algorithm is described and called `BUILD`. In the following subsection we describe this approach.

4.2.1 The Algorithm `BUILD`

In this subsection we review the algorith `BUILD` by Aho et al. [1981a].

This algorithm constructs a phylogenetic tree T on X consistent with a set $\mathscr{R}$ of compatible phylogenetic trees all having leaves in X. We say that $\mathscr{R}$ is *compatible* if $\mathscr{R} = \emptyset$ or if there is an phylogenetic tree T on X that displays every tree contained in $\mathscr{R}$.

The main idea of of the algorithm is to find a partition of the leaf set X according to the trees contained in $\mathscr{R}$. The algorithm will output a tree with a root node whose children are the roots of the trees obtained by recursing on each part of the partition. The base case is when the leaf set contains only one leaf. Then a tree with this single leaf is returned.

The partition of the set of leaves X is performed by `BUILD` using an auxiliary graph that that plays a role in the recursion of the algorithm. This graph is defined as following: let X' be a subset of X, then define the graph $[\mathscr{R}, X']$ with vertex set X'. Put an edge between vertices x and y if there exists a $z \in X'$ and a $\mathscr{T} \in \mathscr{R}$ such that $\mathscr{T}|\{x,y,z\}$ is the restricted subtree in which the path from x to y does not intersect with the path from z to the root. This graph is called the *clustering graph on X' induced by $\mathscr{R}$*. Fig. 4.1(b) shows an example of this graph for where $\mathscr{R} = \{T_1, T_2, T_3, T_4\}$ shown in Fig. 4.1(a), and $X = \{a, b, ..., g\}$.

This approach has been motivated by the following Proposition by Aho et al. [1981b]:

Proposition 3. *If $[\mathscr{R}, X]$ has only one connected component and $|X| > 1$ then $\mathscr{R}$ is not consistent with any phylogenetic tree.*

The key observation here is that for any restricted subtree $\mathscr{T}|\{x,y,z\}$, the leaves labeled by x and y cannot descend from two different children of the root of $\mathscr{T}$, since x and y must belong to the same block [Jansson et al., 2012].

Fig. 4.2 illustrates the output tree of BUILD when applied to the set of trees from Fig. 4.1(a).

Algorithm 1 BUILD($\mathscr{R}$, v, T)

Input : A collection $\mathscr{R}$ of rooted phylogenetic trees and a vertex v.

Output: A rooted phylogenetic tree T that displays $\mathscr{R}$ with root vertex v, or the statement $\mathscr{R}$ is not compatible.

Let $X = x_1, x_2, ..., x_n$ be the label set of $\mathscr{R}$. **if** $|X| = 1$ **then**
  output the rooted phylogenetic tree consisting of the single vertex v labelled by x_1.
end
if $|X| = 2$ **then**
  output the rooted phylogenetic tree consisting of the single vertex v labelled by x_1.
end
if $|X| \geq 3$ **then**
  construct $[\mathscr{R}, X]$.
end
Let $X_1, X_2, ..., X_k$ denote the vertex sets of the components of $[\mathscr{R}, X]$. **if** $k = 1$ **then**
  stop and output $\mathscr{R}$ *is not compatible*.
end
for $i = 1 \rightarrow k$ **do**
  call BUILD$(\mathscr{R}_i, v_i, T_i)$, where $\mathscr{R}_i$ is the collection of rooted phylogenetic trees obtained from $\mathscr{R}$ by restricting each tree in $\mathscr{R}$ to X_i. **if** *BUILD*$(\mathscr{R}_i, v_i, T_i)$ *outputs a tree* **then**
    attach T_i to v via de edge $\{v_i, v\}$.
  end
end

Semple and Steel [2003] have presented the following results:

Theorem 4. *Let $\mathscr{R}$ be a collection of rooted phylogenetic trees. Then* BUILD *applied to $\mathscr{R}$ either*

(i) outputs a rooted phylogenetic tree that displays $\mathscr{R}$ if $\mathscr{R}$ is compatible; or

(ii) outputs the statement "$\mathscr{R}$ is not compatible" otherwise.

In Algorithm 1 we reproduce the description of BUILD. Note that a more efficient solution of the same problem has been described e.g. by Rauch Henzinger et al. [1999], however the authors restrict the set $\mathscr{R}$ to trees that only contain three species.

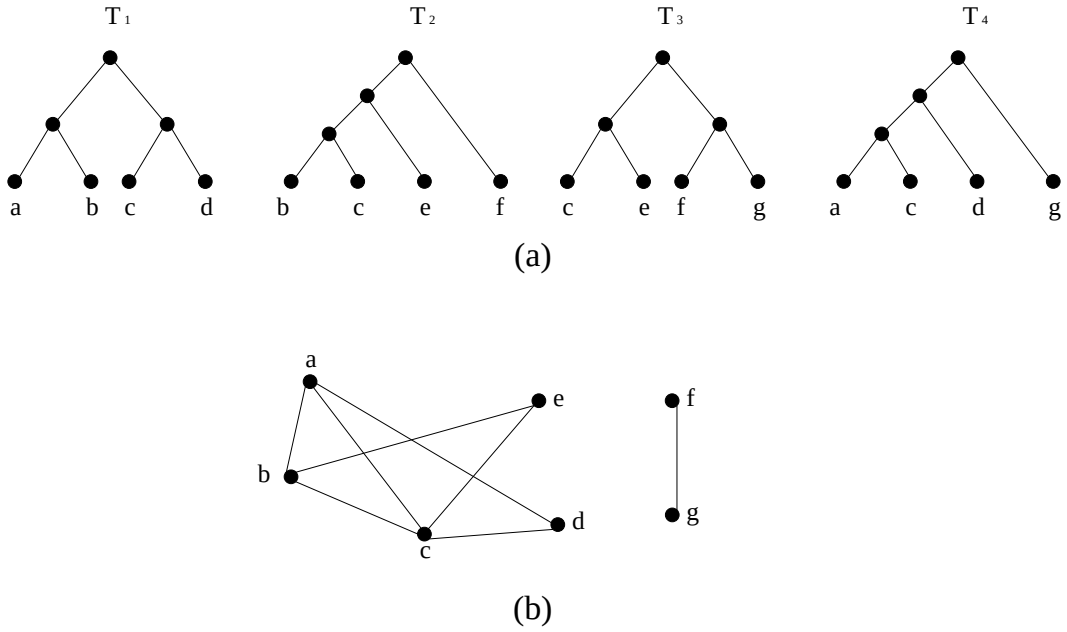


Figure 4.1: (a) The set of phylogenetic trees $\mathscr{R}$. (b) The auxiliary graph $[\mathscr{R}, X]$.

4.2.2 Rooted triples

Rooted trees can be analysed in terms of their smallest phylogenetically informative subtrees called *rooted triples*. In this subsection we will present some previous results related to rooted triples.

The importance of sets of rooted triples stems from the fact that the set $\mathscr{R}_T$ of rooted triples displayed by a phylogenetic tree T uniquely determines T up to isomorphism, i.e. if T' is a phylogenetic tree for which $\mathscr{R}_T = \mathscr{R}_{T'}$ holds then T and T' must be isomorphic. In fact, a more general result of this nature is presented by Semple and Steel [2003]:

Theorem 5. *Let $\mathscr{R}$ be a collection of triples so that the union of their leaf sets is X. Then, when applied* `BUILD` *to $\mathscr{R}$, either:*

(i) outputs a phylogenetic tree on X that displays $\mathscr{R}$ if $\mathscr{R}$ is compatible; or

(ii) outputs the statement "$\mathscr{R}$ is not compatible".

Ng and Wormald [1996] give two algorithms: `ONETREE` and `ALLTREES`, which take a set of rooted triples and fan triples and return a compatible tree and a list of all compatible trees, respectively. In [Bryant and Steel, 1995] a simplification of `ONETREE` is presented, which does not handle fan triples and that has a time complexity of $O(mn)$ for a set of m triples and n leaves.

The requirement that all the triples in the set must be compatible, allows us to infer new phylogenetic relations from the input set and by iteration, we can obtain the *closure* of a set of triples.

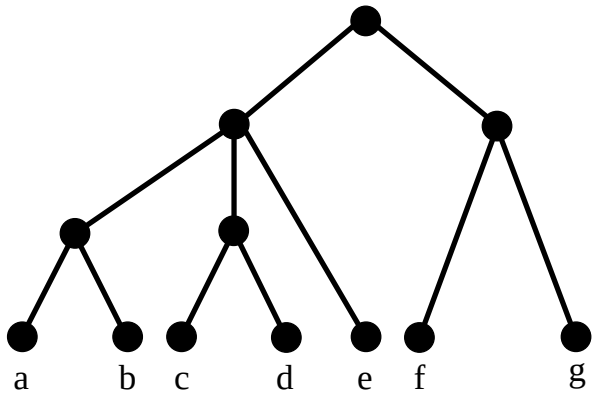


Figure 4.2: The BUILD output tree with $\mathscr{R} = T_1, T_2, T_3, T_4$ from Fig. 4.1(a).

Closure of a set of triples

Given $\mathscr{R}$, a set of of rooted triples, we say $\mathscr{R} \vdash ab|c$ if for each phylogenetic tree that displays $\mathscr{R}$, displays $ab|c$ as well.

Following the notation in [Grünewald et al., 2007], let $co(\mathscr{R})$ be the set of rooted phylogenetic trees on X that display all the triples in $\mathscr{R}$. If $co(\mathscr{R})$ is not empty, then $\mathscr{R}$ is compatible.

The *closure* of $\mathscr{R}$ is defined by

$cl(\mathscr{R}) = \bigcap_{T \in co(\mathscr{R})} \mathscr{R}_T$

where $\mathscr{R}_T$ is the set of all induced triples displayed by T as defined before.

We can as well define the *closure set* as

$cl(\mathscr{R}) = ab|c : \mathscr{R} \vdash ab|c$

The set $\mathscr{R}$ is closed if for each triple $\mathscr{R} \vdash ab|c$ implies that $ab|c \in \mathscr{R}$

This closure operator satisfies the following properties:

- $cl(\mathscr{R})$ is the minimal closed set containing $\mathscr{R}$.
- $cl(\mathscr{R}) = cl(cl(\mathscr{R}))$.
- $\mathscr{R}$ is closed if and only if $\mathscr{R} = cl(\mathscr{R})$.
- T is compatible with $\mathscr{R}$ if and only if T is compatible with $cl(\mathscr{R})$.

The closure operator leads to the following Lemma and Proposition [Grünewald et al., 2007]:

Lemma 6. *Let $\mathscr{R}$ be a set of rooted triples. Then $\mathscr{R}$ is incompatible if and only if there exists a set $\mathscr{R}' \subset \mathscr{R}$ such that for every rooted triple $ab|c \in \mathscr{R} - \mathscr{R}'$ either $\mathscr{R}' \vdash ac|b$ or $\mathscr{R}' \vdash bc|a$.*

For the Proposition we define:

$[\mathscr{R}_1, \mathscr{R}_2] := \{ab|c \in \mathscr{R}_1 : \nexists \mathscr{R}' \subseteq \mathscr{R}_2 : \mathscr{R}' \vdash ac|b \text{ or } \mathscr{R}' \vdash bc|a\}$.

Proposition 7. *Let $\mathscr{R}_1$ and $\mathscr{R}_2$ be two sets of rooted triples (compatible or not) for which $\mathscr{R}_1 \subseteq \mathscr{R}_2$. Then $[\mathscr{R}_1, \mathscr{R}_2]$ is compatible. In particular $[\mathscr{R}_1, \mathscr{R}_1]$ is compatible.*

This proposition is relevant for supertree methods, since we want to ensure that if $yz|x$ is displayed by the output tree, then $yz|x$ is an input triple or implied by some combination of input triples and no input triple or combination of input triples displays or implies $xz|y$ or $xy|z$.

Closure of minimal sets that provides all the information in a tree

Triples play an important role in the branching information of supertrees, in [Rauch Henzinger et al., 1999] it is shown that a phylogenetic tree with n leaves can be represented by a set of $O(n)$ rooted triples, i.e. one rooted triple per edge. Here we show some of the definitions presented in [Grünewald et al., 2007] that lead to this result.

We say that a phylogenetic tree T' refines another phylogenetic tree T if the set of clusters of T is a subset of the set of clusters of T', then we write $T \leqslant T'$.

Definition 8. *A collection of rooted triples $\mathscr{R}$* identifies *a rooted phylogenetic tree T if T displays $\mathscr{R}$ and every other tree that displays $\mathscr{R}$ is a refinement of T.*

Lemma 9. *For any subset $\mathscr{R}$ of $\mathscr{R}_T$, $cl(\mathscr{R}) = \mathscr{R}_T$ if and only if $\mathscr{R}$ identifies T.*

A rooted triple $xy|z$ *distinguishes* an edge e in T if the path from x to z in T intersects the path from y to the root of T only on the edge e.

Corollary 10. *If $\mathscr{R}$ is a minimal set of rooted triples identifying T then each element of $\mathscr{R}$ distinguishes an internal edge of T.*

4.2.3 Inconsistent Set of Triples

Data obtained experimentally often contain errors, meaning that there will be no tree consistent with all the trees in the input set. A single error in one of the input trees will make `BUILD` return the *null* tree. Optimization versions called *the maximum inferred consensus tree problem* (`MICT`) and *the maximum inferred local consensus tree problem* (`MILCT`) have been introduced in [Gasieniec et al., 1999].

`MICT` deals with the construction of a rooted tree that is consistent with as many *LCA constraints* as possible from a given set. An *LCA constraint* on a set X is a constraint of the form $\{i, j\} < \{k, l\}$, where $i, j, k, l \in X$, that specifies that the lowest common ancestor of i and j is a proper descendant of the lowest common ancestor of k and l. If the *LCA constraint* is of the form $\{i, j\} < \{j, k\}$, the constraint is called a *3-leaf constraint* which is the specific case `MILCT` of `MICT` to determine the relative topology of i, j and k.

Jansson [2001] has proved that both `MICT` and `MILCT` are NP-complete. Therefore, other authors have developed approximations to solve this problem [He et al., 2006, Byrka et al., 2010a, Wu, 2004,

Byrka et al., 2010b]. Here we are specifically interested in MILCT since it deals with 3-leaf constraints that represent triples as defined in the previous subsection.

These approximations can be divided in two categories:

- Maximum rooted triples consistency problem (MaxRTC)
- Minimum rooted triples inconsistency problem (MinRTI)

To introduce these problems we need the two following defintions for a phylogenetic tree T over a leaf set X and a set of rooted triples $\mathscr{R}$ over X :

Definition 11. *$J(\mathscr{R},T) = |\mathscr{R} \cap \mathscr{R}_T|$ is the number of rooted triples in $\mathscr{R}$ that are consistent with T.*

Definition 12. *$I(\mathscr{R},T) = |\mathscr{R} \setminus \mathscr{R}_T|$ is the number of rooted triples in $\mathscr{R}$ that are inconsistent with T.*

Maximum rooted triples consistency problem

In [Byrka et al., 2010a] the MaxRTC is defined as following:

Definition 13. *Given a set $\mathscr{R}$ of rooted triples with leaf set X, output a phylogenetic tree T leaf-labeled by X which maximizes $J(\mathscr{R},T)$.*

In [Gasieniec et al., 1999] the greedy top-down algorithms One-Leaf-Split and Min-Cut-Split were presented and were the first polynomial-time approximation algorithms to solve MaxRTC. The idea of One-Leaf-Split is to construct a catepillar tree that is consistent with at least one third of the whole input set of triples. On the other hand, Min-Cut-Split performs similarly to BUILD, with the exception that the edges are weighted in the auxiliar graph and if this forms only one connected component with more than one vertex, then a minimum weight edge cut process is carried out to delete some edges so that the algorithm can continue instead of returning *null*. Semple and Steel [2000] independently developped their heuristic called MinCutSupertree which uses the same idea for merging phylogenetic trees, later Page [2002] presents the modified version of MinCutSupertree to tackle some of the weaknesses of this method.

Snir and Rao [2006] presented an approach called MXC which performs similar to Min-Cut-Split with the difference that instead of deleting edges, MXC adds edges in the auxiliary graph in order to find a cut that maximizes the ratio between the extra edges and the ordinary edges when BUILD is stuck with one connected component.

A different approximation named Best-Pair-Merge-First is presented by Wu [2004] who gives an bottom-up greedy heuristic which runs in polynomial time and makes use of the well known UPGMA/WPGMA and Neighbor-Joining methods [Felsenstein, 2004]. The idea of this approach is to repeatedly merge two sets A and B and then creating a node representing the merged set and whose children are the already existing nodes that represent sets A and B. The algorithm starts with singleton sets, each containing a single leaf label.

Byrka et al. [2010a] present the bottom-up algorithm `Modified-BPMF` which outperforms and is a modified version of Wu's `Best-Pair-Merge-First`. The idea here is to merge two existing trees T_i and T_j whose leaves participate in many rooted triples of the form $xy|z$, where x belongs to T_i and y to T_j but z to none of them. Similar to `One-Leaf-Split`, `Modified-BPMF` guaranties that the output tree is consistent with at least one third of $\mathscr{R}$.

Minimum rooted triples inconsistency problem

Similarly, in [Byrka et al., 2010a] the `MinRTI` is defined as following:

Definition 14. *Given a set $\mathscr{R}$ of rooted triples with leaf set X, output a phylogenetic tree T leaf-labeled by X which minimizes $I(\mathscr{R},T)$.*

As the algorithm `Min-Cut-Split` [Gasieniec et al., 1999] performs the deletion of edges in the auxiliary graph when this is formed by only one connected component, this corresponds to removing one or more rooted triples from $\mathscr{R}$. By introducing a parameter m which is the minimum total weight of triples to remove then this implies that `Min-Cut-Split` gives as well an approximation algorithm for `MinRTI`.

Byrka et al. [2010a] investigate as well approximation algorithms for `MinRTI` that can be used to approximate `MaxRTC`.

The *forbidden rooted triples consistency problem* [He et al., 2006] is a different approach where a set of *good* triples and a set of *forbidden* triples are given, the idea here is to construct a phylogenetic tree that is consistent with the set of all good triples and that is inconsistent with the set of forbidden ones. This approach is motivated by the discovering of some rooted triples that are very unlikely to appear as induced subtrees in the true tree. The algorithm is an extension of `BUILD` that deals with a nonempty set of forbidden rooted triples.

4.3 Minimal Trees

Supertree methods have been criticized because the output tree can yield evolutionary relationships among leaves that are not supported by any of the input trees, which can created novel clades that should be regarded as spurious [Bininda-Emonds, 2004a]. For this reason supertrees containing as few internal nodes as possible while still being consistent with the input trees need to be studied in order to avoid introducing unsupported branching information.

As we have seen so far, most of supertree methods use the same principle as `BUILD`, however in the case where the set of triples is consistent `BUILD` does not always produce a tree with the minimum number of internal nodes. Therefore, for a given triple set $\mathscr{R}$ it does not necessarily generate a minimal phylogenetic tree T that displays $\mathscr{R}$, i.e., T may resolve multifurcations in an arbitrary way that is not implied by any of the triples in $\mathscr{R}$. The problem of constructing a tree consistent with $\mathscr{R}$ and minimizing the number of interior vertices is NP-hard as proved by Jansson et al. [2012].

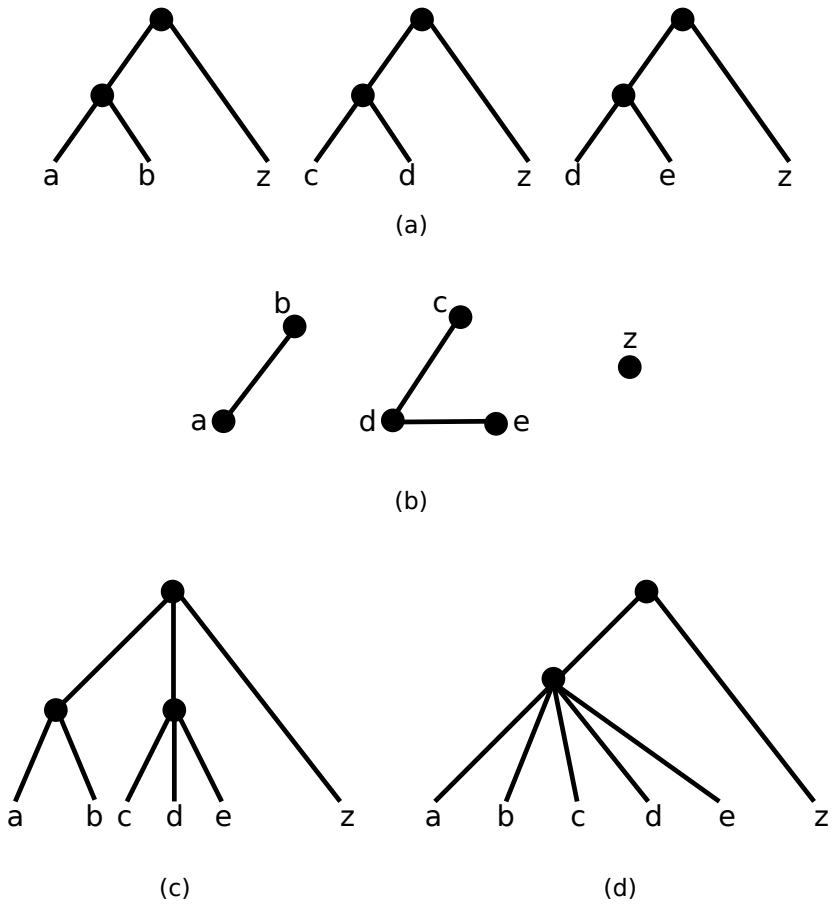


Figure 4.3: **(a)** A consistent set of triples. **(b)** The auxiliary graph retrieved by `BUILD`. **(c)** Output supertree by `BUILD`. **(d)** Minimal resolved supertree.

In fact, they have shown that the output tree produced by `BUILD` might have $\Omega(n)$ times more internal nodes than necessary, where n is the cardinality of X, the leaf set of the input triples. As shown in Fig. 4.3, `BUILD` will produce a tree with two internal nodes consistent with the triple set $\mathscr{R} = \{ab|z, cd|z, de|z\}$. One internal node is the parent of leaves a and b and the other internal node the parent of leaves c, d and e, while the optimal solution is a tree with only one internal node which is the parent of leaves a, b, c, d and e.

Jansson et al. [2012] furthermore, generalize this example by defining $\mathscr{R}$ as follows:

$\mathscr{R} = \{x_1x_2|x_0, x_3x_4|x_0, ..., x_{2i-1}x_{2i}|x_0\}$

This is a set of consistent triples for which `BUILD` will produce a tree with i internal nodes. However, a tree with a root node being the parent of leaf x_0 and the internal node x whose children are the $2i$ leaves $x_1, x_2, ..., x_2i$ is also consistent with $\mathscr{R}$, showing that `BUILD` may produce a tree with Ωn times more internal nodes that the minimally resolved supertree. From this observation the authors define *the minimally resolved supertree consistent with rooted triples problem* (`MinRS`) as following:

Input: A set $\mathscr{R}$ of rooted triples with leaf set X.

Output: A rooted, unordered tree whose leaves are distinctly labeled by X which has as few internal nodes as possible and which is consistent with every rooted triple in $\mathscr{R}$, if such a tree exists; otherwise, *null*.

As this problem has been proved to be NP-hard, the authors provide a modification to `BUILD` to reduce the number of internal nodes. The idea is to compute a minimum coloring on undirected graph and then merge nodes with receive the same color. In this graph each node represents one connected component in the auxiliary graph, and edges are placed between two nodes C_1 and C_2 if $\mathscr{R}$ contains a triple $xy|z$ where $x,y \in C_1$ and $z \in C_2$ or vice versa. However whether this method gives always an optimal solution or not to `MinRS` is still an open question.

Semple [2003] gives an algorithm that produces all minor-minimal trees consistent with $\mathscr{R}$, i.e., if T' is obtained from T by contracting an edge, T' does not display $\mathscr{R}$ anymore. He also proves that the tree produced by `BUILD` is minor-minimal. However, depending on $\mathscr{R}$, not all trees consistent with $\mathscr{R}$ can be obtained from `BUILD`.

The definition of a *minimal* tree in this context is a rooted phylogenetic tree T that is consistent with a set of triples $\mathscr{R}$ if no internal edge of T can be contracted so that the resulting tree is also consistent with $\mathscr{R}$. The purpose is to find the set $\mathscr{T}_{\mathscr{R}}^{min}$ of all minimal trees consistent with $\mathscr{R}$. The motivation for finding $\mathscr{T}_{\mathscr{R}}^{min}$ is that it contains all the information provided by the set of all phylogenetic trees consistent with $\mathscr{R}$ since any other tree consistent with $\mathscr{R}$ that is not minimal can be deduced from $\mathscr{T}_{\mathscr{R}}^{min}$ by resolving internal nodes of some tree in from $\mathscr{T}_{\mathscr{R}}^{min}$.

Semple [2003] gives an algorithm called `AllMinTrees` that requires only polynomial time for the calculation of each of the possibly exponentially many minor-minimal trees.

4.4 Symbolic Ultrametrics and the Link to Phylogenetic Trees

It turned out that there is a 1-to-1 correspondance between symbolic ultrametric and phylogenetic trees when these are defined as symbolically dated trees. Ultrametrics are well-studied in phylogenetics as they correspond to weighted, rooted trees.

In this section, we recall some results from Böcker and Dress [1998] and Semple and Steel [2003] concerning symbolic ultrametrics and their relation to phylogenetic trees.

Suppose that $T = (V,E)$ is a phylogenetic tree on X and that $t : V \to M^{\odot}$ is a map such that $t(x) = \odot$ for all $x \in X$. We call such a map t a *symbolic dating map* for T; it is *discriminating* if $t(u) \neq t(v)$, for all edges $\{u,v\} \in E$. To the pair $(T;t)$ we associate the map $d_{(T;t)}$ on $X \times X$ by setting, for all $x,y \in X$,

$$d_{(T;t)} : X \times X \to M^{\odot};\ d_{(T;t)}(x,y) = t(\mathrm{lca}_T(x,y)). \tag{4.1}$$

Clearly this map is symmetric and satisfies (U0). We call the pair $(T;t)$ a *symbolic representation* of a map $\delta : X \times X \to M^{\odot}$ if $\delta(x,y) = d_{(T;t)}(x,y)$ holds for all $x,y \in X$; it is called discriminating if t is

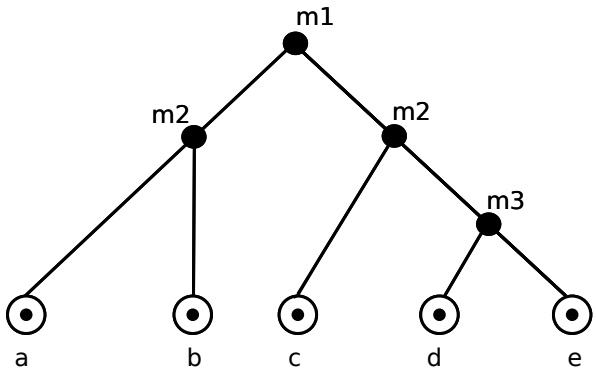


Figure 4.4: A phylogenetic tree $T = (V,E)$ on the set $X = \{a,\ldots,e\}$, together with a map t from the set of interior vertices of T to the set of events $M = \{m_1,m_2,m_3\}$, as indicated by the labels on the interior vertices of T. The vertex in V that is the least common ancestor of c and e has label m_2 and so $d_{(T;t)}(c,e) = m_2$.

discriminating (see Fig. 4.4 for an example of a discriminating symbolic representation). Note that we call two symbolic representations $(T;t)$ and $(T';t')$ of δ *isomorphic* if T and T' are isomorphic via a map $\psi : V(T) \to V(T')$ such that $t'(\psi(v)) = t(v)$ holds for all $v \in V(T)$.

In [Böcker and Dress, 1998], the following fundamental results concerning the relationship between symbolic ultrametrics and symbolic representations are proven:

Theorem 15. *Suppose $\delta : X \times X \to M^{\odot}$ is a map. Then there is a discriminating symbolic representation $\delta(T;t)$ if and only if $\delta(T;t)$ is a symbolic ultrametric. Furthermore, up to isomorphism, this representation is unique.*

Given any symbolic ultrametric δ on X, we denote the unique discriminating symbolic representation of δ by $(T_\delta;t_\delta)$, $t_\delta(u) = \delta(x,y)$ for every pair of leaves x,y in $T(u)$.

Another very importante result from the same authors that is related to rooted triples and supertrees is stated in the following Theorem:

Theorem 16. *Given the finite set X, there exists a canonical 1-to-1 correspondence between*

(i) (isomomorphism classes of) rooted trees $T = (V,E)$ with leaf set X

(ii) ternary relations $\int$ defined on X (with $(x,y,z) \in \int$ denoted by $xy\int z$, for $x,y,z \in X$) satisfying the following assertions for all $x,y,z,w \in X$

(H1) $xx\int y \Leftrightarrow x \neq y$

(H2) $xy\int z \Rightarrow yx\int z$

(H3) $xy\int w$ and $yz\int w \Rightarrow xz\int w$

(H4) $xy\int z$ and $yz\int w \Rightarrow xz\int w$.

This correspondance is given by $xy \int z$ if and only if $lca(x,y) \neq lca(x,z) = lca(y,z)$, for all $x,y,z \in X$ which makes the ternary relation satisfy (H1) - (H4). One can immediately observe that this has a direct relation with discriminating symbolic representation for phylogenetic trees.

Now we are ready to present the following relevant result concerning triples and ultrametrics:

Lemma 17. *Given a discriminating symbolic representation $(T_\delta; t_\delta)$, let the map δ denote the induced ultrametric. Let $\int$ denote the ternary relation on X induced by $T = (V,E)$. Then, for all $x,y,z \in X$ we have $xy \int z$ if and only if one of the two following conditions holds:*

- $\delta(x,y) \neq \delta(x,z) = \delta(y,z)$, *or*
- $\delta(x,y) = \delta(x,z) = \delta(y,z)$, *and there is some $w \in X$ with $\delta(x,w) = \delta(y,w) \neq \delta(z,w) = \delta(x,y)$.*

In particular, the relation $\int$ can be recovered and therefore T up to isomorphism from δ.

4.5 Concluding Remarks

We have presented definition and properties of phylogenetic trees. Moreover, we have analysed a specific type of phylogenetic tree called "triple", a phylogenetic tree in three leaves that has been shown to contain much of the information of the phylogenetic tree that displays them. Furthermore, we have presented methods for supertree reconstruction that use triple sets for the reconstruction of phylogenetic trees. Finally, we have presented some results from Böcker and Dress [1998] that throw light on the relation between symbolic ultrametrics and phylogenetic trees. These properties show that this type of trees can be characterized mathematically in terms of symbolic representations. In the following chapters, we will make use of the terminology here presented.

CHAPTER 5

Orthology Relations, Symbolic Ultrametrics, and Cographs

Orthology detection is an important problem in comparative and evolutionary genomics, consequently, a variety of orthology detection methods have been devised in recent years. Although many of these methods are dependent on generating gene and/or species trees, as described in Chapter 3, it has been shown that orthology can be estimated at acceptable levels of accuracy without having to infer gene trees and/or reconciling gene trees with species trees. Thus, it is of interest to understand how much information about the gene tree, the species tree, and their reconciliation is already contained in the orthology relation on the underlying set of genes.

Here we shall show that a result by Böcker and Dress [1998] concerning symbolic ultrametrics, and subsequent algorithmic results by Semple and Steel [2003] for processing these structures can throw a considerable amount of light on this problem. We also prove some mild generalizations of these results that are relevant when dealing with orthology relations. More specifically, we present some new characterizations for symbolic ultrametrics. In so doing we shall also show that, somewhat surprisingly, symbolic ultrametrics are very closely related to cographs, graphs that do not contain an induced path on any subset of four vertices. We also show that the tree corresponding to a symbolic ultrametric can also be recovered using cotrees, trees that can be canonically associated to cographs.

We conclude with some remarks on how these results might be applied in practice to orthology detection.

5.1 Orthology Relations

Suppose that X is a set of genes having a common origin, and that their evolutionary history is given by a gene tree, i.e. a (graph-theoretical) tree $T = (V, E)$ with vertex set V, edge set E and leaf set X.

Typically one can think of T as being derived from a species tree, in which case the interior vertices of T will correspond to speciation or duplication events. In reality, other events such as horizontal gene transfer might also occur, although we will not consider these explicitly here.

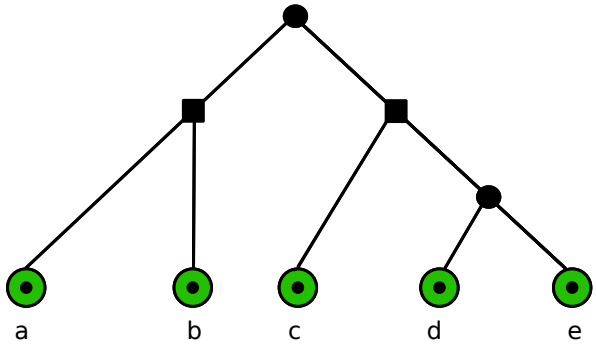


Figure 5.1: A phylogenetic tree $T = (V,E)$ on the set $X = \{a,\ldots,e\}$, together with a map t from the set of interior vertices of T to the set of events $M = \{\bullet, \blacksquare\}$. Leaves are depicted by the symbol $\odot$.

Two genes x,y in X are orthologs if the event corresponding to the (unique) least common ancestor $\mathrm{lca}_T(x,y)$ of x and y in T is a speciation; if x and y are not orthologs then $\mathrm{lca}_T(x,y)$ will correspond to some other events such as a duplication. In particular, we obtain a map t from the set of interior vertices of T to some set M of events, and, consequently, a map $d_{(T;t)}$ from distinct pairs x,y in X to M given by putting $d_{(T;t)}(x,y) = t(\mathrm{lca}_T(x,y))$. These concepts are illustrated in Fig. 5.1. Note that in practice, we do not necessarily know the pair $(T;t)$, but that there are bioinformatics methods that allow us to estimate the values $d_{(T;t)}(x,y)$ for $x,y \in X$ [Altenhoff and Dessimoz, 2009, Lechner et al., 2011]. The question that interests us here can be stated as follows:

Given an arbitrary symmetric map $\delta : X \times X \to M$, i.e. an orthology relation, can we determine if there is a pair $(T;t)$ for which $d_{(T;t)}(x,y) = \delta(x,y)$ holds for $x,y \in X$ distinct and, if not, can we at least find some pair $(T;t)$ where this is almost true?

In the following sections we develop an answer to this question.

5.2 Symbolic Ultrametrics

Suppose that $T = (V,E)$ is a phylogenetic tree on X and that $t : V \to M^{\odot}$ is a symbolic dating map. Note that the symbolic tree representation of an orthology relation on a set of genes need not necessarily be discriminating, since duplication events do not necessarily have to come directly after speciation events and vice versa. To help deal with this, we shall now prove a simple result concerning the relationship between symbolic ultrametrics and arbitrary symbolic representations. To this end, suppose that t is then not discriminating. Then there must exist some $e = \{u,v\} \in E^0$ such that $t(u) = t(v)$.

Let v_e denote the vertex in T obtained by collapsing the edge e. Then the tree $T_e = (V_e, E_e)$ with vertex set $V_e = V \setminus \{u,v\} \cup \{v_e\}$, edge set $E_e = E \setminus \{e\} \cup \{\{e_v, w\} : \{w,u\} \text{ or } \{w,v\} \in E\}$ is clearly a phylogenetic tree on X. Furthermore the map $t_e : V_e \to M^{\odot}$ defined by putting, for all $w \in V_e$,

$$t_e(w) = t(w) \text{ if } w \neq v_e \text{ and } t(v_e) = t(u) \tag{5.1}$$

is again a symbolic dating map for T_e. Clearly, this construction can be repeated, with $(T_e;t_e)$ now playing the role of $(T;t)$, until a phylogenetic tree $\hat{T} = (\hat{V}, \hat{E})$ on X is obtained together with a discriminating symbolic dating map $\hat{t}$ on $\hat{T}$.

Proposition 18. *Let $\delta : X \times X \to M^{\odot}$ be a map. Then the following are equivalent:*

(i) δ is a symbolic ultrametric.

(ii) there is a discriminating symbolic representation of δ.

(iii) there is a symbolic representation of δ.

Moreover, if δ is a symbolic ultrametric, and $(T;t)$ is any symbolic representation of δ, then $(\hat{T};\hat{t})$ is isomorphic to $(T_\delta;t_\delta)$.

Proof. (i) $\Rightarrow$ (ii): Apply Theorem 15 (previous chapter).

(ii) $\Rightarrow$ (iii): This is obvious.

(iii) $\Rightarrow$ (i): It is straight-forward to check that if there is a symbolic representation $(T;t)$ of δ, then δ must satisfy (U0)–(U3). Then apply Theorem 15.

To see that the final statement holds, note that if $\delta : X \times X \to M^{\odot}$ has a symbolic representation $(T;t)$, and $e = \{u,v\} \in E(T)$ with $t(u) = t(v)$, then $d_{(T_e;t_e)} = d_{(T;t)}$. Therefore, $d_{(\hat{T};\hat{t})} = d_{(T';t')}$ must also hold. Moreover, $\hat{t}$ is discriminating by construction and thus, by Theorem 15, the proposition follows. □

We conclude this section by recalling a practical approach for constructing the discriminating symbolic representation $(T_\delta;t_\delta)$ from a given symbolic ultrametric $\delta : X \times X \to M^{\odot}$ based on the algorithm `BUILD`.

This result is a mild generalization from [Semple and Steel, 2003, p. 167-8] which states the following:

Theorem 19. *Let M be a finite set, and let δ be a map from $X \times X$ into M. Then, there is a discriminating symbolic representation of δ if and only if δ is a symbolic ultrametric. Furthermore, up to isomorphism, this representation is unique.*

Before presenting our generalization stated in proposition 20 we introduce the following definitions. Let $\delta : X \times X \to M^{\odot}$ be a symbolic ultrametric on X and let $\mathscr{R}(\delta)$ be the set of triples $xy|z$, $\{x,y,z\} \in \binom{X}{3}$ satisfying one of the following two conditions:

(R1) $\delta(x,y) \neq \delta(x,z) = \delta(y,z)$, or

(R2) $\delta(x,y) = \delta(x,z) = \delta(y,z)$, and there is some $w \in X$ such that $\delta(x,w) = \delta(y,w) \neq \delta(z,w) = \delta(x,y)$.

Furthermore, denote by $\mathscr{R}_\delta \subseteq \mathscr{R}(\delta)$ the subset of $\mathscr{R}(\delta)$ consisting only of the triples satisfying condition (R1). If δ is a symbolic ultrametric then $\mathscr{R}(\delta) = \mathscr{R}_{T_\delta}$ (Lemma 17).

Proposition 20. *Let $\delta : X \times X \to M^\odot$ be a map that satisfies Properties (U0)–(U2). Then the following are equivalent:*

(i) δ is a symbolic ultrametric.

(ii) $\mathscr{R}(\delta)$ is compatible.

(iii) $\mathscr{R}_\delta$ is compatible.

In particular, δ is a symbolic ultrametric if and only if the `BUILD` *algorithm applied to $\mathscr{R}_\delta$ or $\mathscr{R}(\delta)$ returns a phylogenetic tree T, in which case the map $t : V(T) \to M^\odot$, $v \mapsto \delta(x,y)$ with $v = \mathrm{lca}_T(x,y)$, $x,y \in X$, is well-defined and $(T;t)$ is isomorphic to the discriminating symbolic representation for δ.*

Proof. Clearly all 3 assertions are equivalent if $|X| = 3$. So assume $|X| \geq 4$. The implications (i) $\Rightarrow$ (ii) and (ii) $\Rightarrow$ (iii) are trivial in view of the observation preceding Proposition 20.

(iii) $\Rightarrow$ (i): Suppose for contradiction that $\mathscr{R}_\delta$ is compatible but that δ is not a symbolic ultrametric. Then δ does not satisfy Property (U3) and so there exists some $\{x,y,u,v\} \in \binom{X}{4}$ such that $\delta(x,y) = \delta(y,u) = \delta(u,v) \neq \delta(y,v) = \delta(x,v) = \delta(x,u)$. But then $\mathscr{R} := \{xy|v, xu|y, uv|x\} \subseteq \mathscr{R}_\delta$ must hold which is impossible as $\mathscr{R}$ is not compatible and thus $\mathscr{R}_\delta$ cannot be compatible. □

It follows from this result and Theorem 20 that we can decide in polynomial time whether or not δ is a symbolic ultrametric by applying the `BUILD` algorithm to the set $\mathscr{R}_\delta$, which will also construct a symbolic representation of δ in case it is. The following additional consequence, which will not be used later, is also worth noting:

Corollary 21. *Suppose δ is a symbolic ultrametric on X. Then δ has a unique symbolic representation if and only if $|\mathscr{R}(\delta)| = \binom{|X|}{3}$.*

Proof. Suppose first that $|\mathscr{R}(\delta)| = \binom{|X|}{3}$. Then $|\mathscr{R}_{T_\delta}| = \binom{|X|}{3}$ in view of Lemma 17 recalled above as δ is a symbolic ultrametric. Since only a binary phylogenetic tree can display $\binom{|X|}{3}$ triples, it follows that T_δ must be binary. But this implies immediately that $(T_\delta; t_\delta)$ is the unique symbolic representation for δ because any symbolic representation for δ can be obtained from $(T_\delta; t_\delta)$ by resolving interior vertices of T_δ.

Conversely, assume that δ has a unique symbolic representation $(T;t)$. Then T must be binary as otherwise, by Proposition 18, there would exist an interior vertex of T that could be resolved to obtain a new symbolic representation $(T';t')$ for δ contradicting the uniqueness of $(T;t)$. But then $(T;t)$ is

isomorphic to $(T_\delta; t_\delta)$ and so $|\mathscr{R}_{T_\delta}| = \binom{|X|}{3}$. Since δ is a symbolic ultrametric on X, Lemma 17 implies $\mathscr{R}_{T_\delta} = \mathscr{R}(\delta)$ and so the corollary follows. □

It follows from this corollary that if the tree T_δ is unique then it must be fully resolved and there is a discriminating symbolic representation for δ since for each triple in $\mathscr{R}(\delta)$ which is displayed by T_δ, there is a discrimitating symbolic representation of δ as well.

5.3 Cographs and Cotrees

In this section, we investigate a connection between symbolic ultrametrics and *complement-reducible graphs* or *cographs*. See section 2.5 for more material on cographs.

Let $\delta : X \times X \to M^\odot$ be a map satisfying Properties (U0) and (U1) of Definition 2.2. For $x \in X$ and $m \in M$, we define the *neighborhood* $N_m(x)$ of x with respect to m and δ as

$$N_m(x) = N_{m,\delta}(x) := \{y \in X : \delta(x,y) = m\}. \tag{5.2}$$

Note that, in view of Property (U0), $x \notin N_m(x)$ and that, in view of Property (U1), $y \in N_m(x)$ if and only if $x \in N_m(y)$. We also define, for each fixed $m \in M$, an undirected graph $G_m(\delta) = (V_m, E_m)$ with vertex set $V_m = V_m(\delta) = X$ and edge set

$$E_m = E_m(\delta) := \left\{ \{x,y\} \in \binom{X}{2} : y \in N_m(x),\ x \in X \right\}. \tag{5.3}$$

For example, if $\delta = d_{(T;t)}$ for the pair $(T;t)$ depicted in Fig. 5.2(b), then $G_{m_1}(\delta)$ is the graph with vertex set $\{a,\ldots,e\}$ and edge set $\{\{a,d\},\{a,c\},\{a,e\},\{b,d\},\{b,c\},\{b,e\}\}$, $G_{m_2}(\delta)$ and $G_{m_3}(\delta)$ are the graphs with the same vertex set as $G_{m_1}(\delta)$ and edge set $\{\{a,b\},\{c,d\},\{c,e\}\}$ and $\{\{d,e\}\}$, respectively. The graph $G_m(\delta) = (V_m, E_m)$ is shown in Fig. 5.2(a).

In the following result we present the aforementioned connection between symbolic ultrametrics and cographs:

Proposition 22. *Let $\delta : X \times X \to M^\odot$ be a map satisfying Properties (U0) and (U1). Then δ is a symbolic ultrametric if and only if*

(U2') *For all $\{x,y,z\} \in \binom{X}{3}$ there is an $m \in M$ such that $E_m(\delta)$ contains two of the three edges $\{x,y\}$, $\{x,z\}$, and $\{y,z\}$.*

(U3') *$G_m(\delta)$ is a cograph for all $m \in M$.*

Proof. Suppose that δ is a map as in the statement of the proposition. Note that we may assume $|X| \geq 4$.

Clearly, δ satisfies (U2) if and only if it satisfies Property (U2'). Moreover, it is easy to see that (U3') implies (U3). Thus if (U3') and (U2') hold, then δ is a symbolic ultrametric on X. Thus, it only

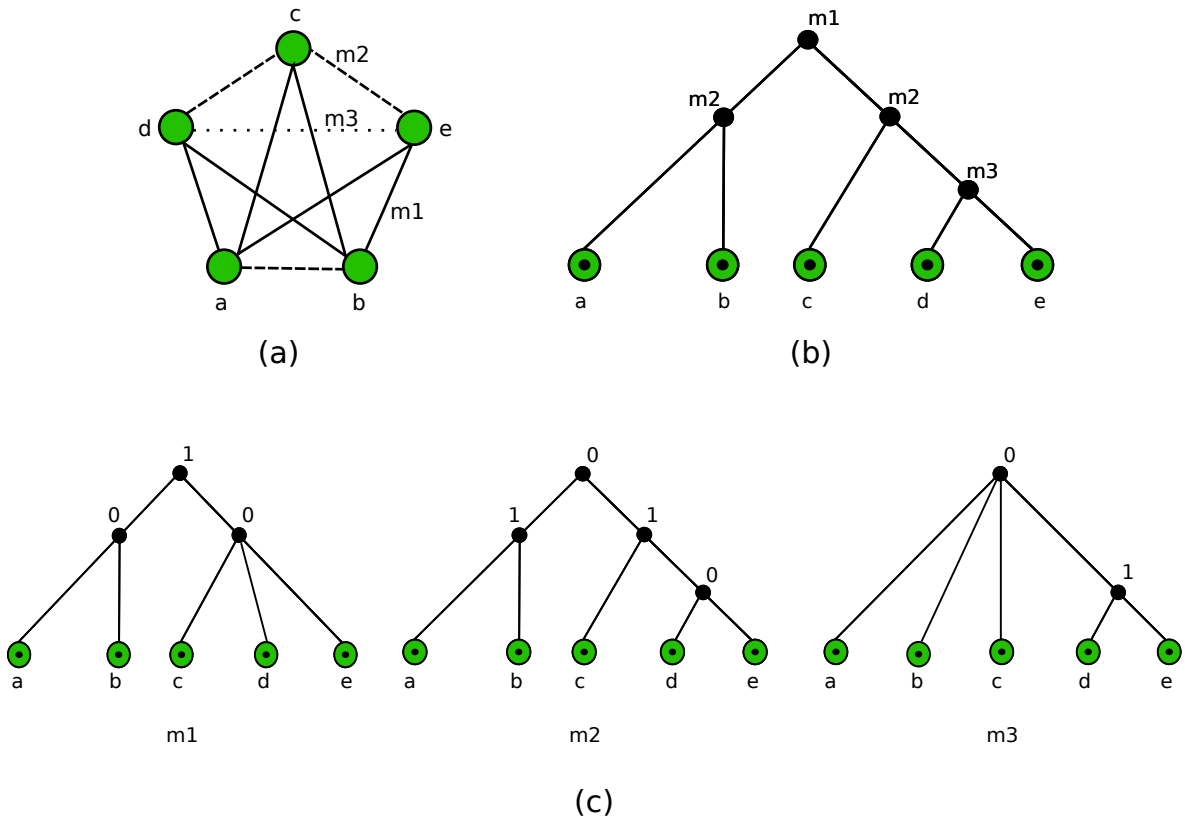


Figure 5.2: For the symbolic ultrametric $\delta = d_{(T;t)}$, with $(T;t)$ pictured in (b), the three cotrees $(T(G_{m_i}(\delta)), \lambda_{G_{m_i}(\delta)})$, $i = 1,2,3$, pictured in that order from left to right in (c). Note that the tree T depicted in (b) refines each of the cotrees. The corresponding $G_{m_i}(\delta)$ is depicted in (a).

remains to show that if δ satisfies (U2) and (U3) (i.e. δ is a symbolic ultrametric), then it must satisfy (U3').

Suppose this is not the case, i.e. (U3') does not hold. Then there exists $\{x,y,u,v\} \in \binom{X}{4}$ and some $m \in M$ such that the subgraph of $G(\delta)$ induced on $\{x,y,u,v\}$ is a path of length three. Suppose that this path is x,y,u,v. Then $\delta(x,y) = \delta(y,u) = \delta(u,v) = m$ and $m \notin \{\delta(x,u), \delta(x,v), \delta(y,v)\}$. But (U2) implies $\delta(x,u) = \delta(x,v) = \delta(y,v)$, and so (U3) does not hold. This contradiction completes the proof. □

Intriguingly, it is well-known in the literature concerning cographs that, to any cograph G, one can associate a canonical *cotree* $T(G) = (V,E)$. This is a rooted tree with root[1] ρ, leaf set equal to the vertex set $V(G)$ of G and inner vertices that represent so-called "join" and "union" operations together with a labeling map $\lambda_G : V^0 \to \{0,1\}$ such that $\lambda_G(\rho) = 1$ and, if $v \in V^0$ and $w_1, \ldots, w_k \in V^0$, $k \geq 2$, are the children of v, then $|\lambda_G(v) - \lambda_G(w_i)| = 1$, for all $i = 1, \ldots, k$ (cf. [Corneil et al., 1981]). For

[1] Note that in cotrees the root might have outdegree one; in such cases we will simply suppress this vertex and its outgoing edge.

example, if $\delta = d_{(T;t)}$ for the pair $(T;t)$ depicted in Fig. 5.2(b), then the cotrees associated to the cographs $G_{m_1}(\delta)$, $G_{m_2}(\delta)$, and $G_{m_3}(\delta)$, respectively, are depicted in Fig. 5.2(c). Note that the cotree associated to a cograph has root labeled with 0 if and only if the cograph is disconnected.

The key observation about cographs is that, given a cograph G, a pair $\{x,y\} \in \binom{V(G)}{2}$ is an edge in G if and only if $\lambda_G(\mathrm{lca}_{T(G)}(x,y)) = 1$ (cf. [Corneil et al., 1981, p. 166]). It is therefore natural to ask what the relationship is between the discriminating representation of a symbolic ultrametric δ and the cotrees associated to the cographs coming from δ given by Proposition 22. We shall now show that there is a very close connection between these structures.

To this end, suppose $\delta : X \times X \to M^{\odot}$ is a map satisfying Properties (U0) and (U1) and $m \in M$. Consider the map $\delta_m : X \times X \to \{0,1,\odot\}$ defined, for all $x,y \in X$, by putting

$$\delta_m(x,y) = \begin{cases} \odot & \text{if} \quad x = y, \\ 1 & \text{if} \quad \{x,y\} \in E_m(\delta), \\ 0 & \text{if} \quad \text{else.} \end{cases} \tag{5.4}$$

Note if δ is a symbolic ultrametric on X, then it is easy to see that δ_m is also a symbolic ultrametric on X, $m \in M$ (essentially because $G(\delta_m)$ is a cograph).

Lemma 23. *Let $\delta : X \times X \to M^{\odot}$ be a symbolic ultrametric. Then, for all $m \in M$, $(T(G_m(\delta)); \lambda_{G_m(\delta)})$ is the discriminating symbolic representation for δ_m.*

Proof. Suppose $m \in M$, and let $T' = T(G_m(\delta))$ and $t' = \lambda_{G_m(\delta)}$. In view of Theorem 15 it suffices to show that $\delta_m(x,y) = d_{(T';t')}(x,y)$ holds for all $x,y \in X$. Let $x,y \in X$. Then, by the aforementioned properties of the cotree associated to a cograph and Proposition 22, it follows that $d_{(T';t')}(x,y) = t'(\mathrm{lca}_{T'}(x,y)) = 1$ if and only if $\{x,y\} \in E_m(\delta)$ if and only if $\delta_m(x,y) = 1$, as required. □

Using this lemma, we now prove a technical result which, given a symbolic ultrametric δ, relates triples in $\mathscr{R}(\delta)$ and, for $m \in M$, triples in $\mathscr{R}_{\delta_m}$.

Theorem 24. *Let $\delta : X \times X \to M^{\odot}$ be a symbolic ultrametric. Then the following holds:*

(i) For all $m \in M$, $\mathscr{R}_{\delta_m} \subseteq \mathscr{R}_{\delta}$.

(ii) For all $m \in M$, $\mathscr{R}(\delta_m) \subseteq \mathscr{R}(\delta)$.

(iii) $\mathscr{R}_{\delta} = \bigcup_{m \in M} \mathscr{R}_{\delta_m}$.

Proof. (i) Suppose $m \in M$ and $xy|z \in \mathscr{R}_{\delta_m}$. Then $\delta_m(x,y) \neq \delta_m(x,z) = \delta_m(y,z)$ and so either (a) $\delta_m(x,y) = 1$ and $\delta_m(x,z) = \delta_m(y,z) = 0$ or (b) $\delta_m(x,y) = 0$ and $\delta_m(x,z) = \delta_m(y,z) = 1$.

If Case (a) holds then $\{x,y\} \in E_m(\delta)$ and $\{x,z\}, \{y,z\} \notin E_m(\delta)$. Hence $\delta(x,y) = m$ and $\delta(x,z), \delta(y,z) \neq m$. Since δ is an ultrametric and so satisfies Property (U2) it follows that $\delta(x,z) = \delta(y,z)$. Consequently, $xy|z \in R_{\delta}$ in this case.

If Case (b) holds then $\{x,y\} \notin E_m(\delta)$ and $\{x,z\},\{y,z\} \in E_m(\delta)$. But then $\delta(x,z) = \delta(y,z) = m \neq \delta(x,y)$ and so $xy|z \in R_\delta$ must hold in this case, too.

(ii) Let $m \in M$. Suppose $xy|z \in \mathscr{R}(\delta_m)$. Assume first that $xy|z$ satisfies Property (R1). Then Assertion (i) implies $xy|z \in \mathscr{R}_\delta \subseteq \mathscr{R}(\delta)$. So assume that $xy|z$ does not satisfy Property (R1). Then $xy|z \notin \mathscr{R}_{\delta_m}$ and $xy|z$ must satisfy Property (R2), that is, $\delta_m(x,y) = \delta_m(x,z) = \delta_m(y,z)$ and there must exist some $w \in X$ such that $\delta_m(x,w) = \delta_m(y,w) \neq \delta_m(z,w) = \delta_m(x,y)$. We distinguish the cases $\delta_m(x,y) = \delta_m(x,z) = \delta_m(y,z) = 1$ and $\delta_m(x,y) = \delta_m(x,z) = \delta_m(y,z) = 0$.

Assume first that $\delta_m(x,y) = \delta_m(x,z) = \delta_m(y,z) = 1$ holds. Then $m = \delta(x,y) = \delta(x,z) = \delta(y,z)$ and so $\delta(z,w) = m$ and $m \notin \{\delta(x,w),\delta(y,w)\}$ must hold. But then Property (U2) implies that $\delta(x,w) = \delta(y,w) \neq m$ and so (R2) holds. Thus, $xy|z \in \mathscr{R}(\delta)$ in this case.

Now, assume that $\delta_m(x,y) = \delta_m(x,z) = \delta_m(y,z) = 0$ holds. Then $m \notin \{\delta(x,y),\delta(x,z),\delta(y,z),\delta(z,w)\}$ and so $m = \delta(x,w) = \delta(y,w)$. By Property (U2) it follows that $m_1 := \delta(y,z) = \delta(z,w) = \delta(z,x) \neq m$. If $m_2 := \delta(x,y) = m_1$ then $xy|z$ satisfies Property (R2) for δ and so $xy|z \in \mathscr{R}(\delta)$. If $m_2 \neq m_1$ then $xy|z \in \mathscr{R}_{\delta_{m_2}} \subseteq \mathscr{R}_\delta \subseteq \mathscr{R}(\delta)$ in view of Assertion (i). This completes the proof of (ii).

(iii) Statement (i) clearly implies $\bigcup_{m\in M} \mathscr{R}_{\delta_m} \subseteq \mathscr{R}_\delta$. To see that the converse set inclusion holds, let $xy|z \in \mathscr{R}_\delta$. Then there exists some $m \in M$ such that $m = \delta(x,y) \neq \delta(x,z) = \delta(y,z)$ and thus $\{x,y\} \in E_m(\delta)$ and $\{x,z\},\{y,z\} \notin E_m(\delta)$. Hence, $\delta_m(x,y) = 1 \neq 0 = \delta_m(x,z) = \delta_m(y,z)$ and so $xy|z \in \mathscr{R}_{\delta_m}$, as required. □

Using this theorem, we now see how the discriminating symbolic representation T_δ of a symbolic ultrametric δ can be constructed from the cotrees $T(G_m(\delta))$, $m \in M$ (or, equivalently, the discriminating symbolic representations of the maps δ_m, $m \in M$). The first statement of the following corollary is illustrated in Fig. 5.2(c).

Corollary 25. *Let $\delta : X \times X \to M^\odot$ be a symbolic ultrametric. Then, for each $m \in M$, $T(G_m(\delta)) \leq T_\delta$. Moreover, T_δ is isomorphic to the tree obtained by applying* `BUILD` *to the set $\bigcup_{m\in M} \mathscr{R}_{\delta_m}$.*

Proof. The second statement follows immediately from Theorem 24(ii) and Proposition 20.

To see that $T(G_m(\delta)) \leq T_\delta$ holds for all $m \in M$, note that since δ_m is a symbolic ultrametric $\mathscr{R}(\delta_m) = \mathscr{R}_{T_{\delta_m}}$ holds by Lemma 2 of [Böcker and Dress, 1998] recalled above. Hence by Theorem 24 (ii), $\mathscr{R}_{T_{\delta_m}} \subseteq \mathscr{R}_{T_\delta}$. By Theorem 6.4.1 of [Semple and Steel, 2003] this last statement holds if and only if $T_{\delta_m} \leq T_\delta$. Now apply Lemma 23. □

By modifying the argument in the proof of part (iii) of Theorem 24, it is straight-forward to show, under the same assumptions given in the theorem plus the additional assumption $|M| \geq 3$, that T_δ is isomorphic to the tree obtained by applying `BUILD` to the set $\bigcup_{m\in M'} \mathscr{R}_{\delta_m}$, for any $M' \subseteq M$ with $|M'| = |M| - 1$. However, in general it is not possible to obtain T_δ using `BUILD` in this way by using subsets of M with size less than $|M| - 1$ (see Fig. 5.3).

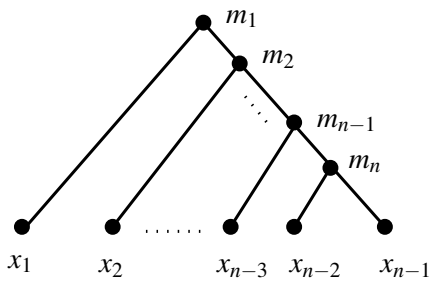


Figure 5.3: A symbolic representation of a symbolic ultrametric δ on the set $X = \{x_1, \ldots, x_{n-1}\}$ with values in the set $M = \{m_1, \ldots, m_n\}$. It can be shown that it is not possible to reconstruct T_δ by applying BUILD to the set $\bigcup_{m \in M'} R_{\delta_m}$, for any $M' \subseteq M$ with $|M'| \leq n-2$.

5.4 Concluding Remarks

The case of most immediate practical relevance for the results presented in this chapter is the case $|M| = 2$, where the events in M are simply speciation and duplication. Here, we assume that we are given an arbitrary orthology relation $\delta : X \times X \to \{0,1\}^\odot$ on a set X of genes (i.e., a map that satisfies (U0) and (U1) and that assigns the value 1 to pairs of genes that are (co-)orthologs and 0 to pairs that are paralogs), a relation that can be reliably estimated from X using various bioinformatics techniques; cf. e.g. [Lechner et al., 2011] and the reference therein. We then aim to obtain a symbolic representation $(T;t)$ of δ, such that $x, y \in X$ are orthologs if $\mathrm{lca}_T(x,y)$ corresponds to a speciation event and paralogs if $\mathrm{lca}_T(x,y)$ corresponds to a duplication event within a single lineage (i.e. $t(\mathrm{lca}_T(x,y))$ equals 1 or 0, respectively).

The above results immediately provide the following characterizations of orthology relations for which a symbolic representation exists:

Corollary 26. *Suppose that $\delta : X \times X \to \{0,1\}^\odot$ is an orthology relation. Then the following are equivalent:*

(i) *δ has a symbolic representation.*

(ii) *δ is a symbolic ultrametric.*

(iii) *$G_1(\delta) = \overline{G_0(\delta)}$ is a cograph.*

Somewhat surprisingly, this simple characterization of "ideal" orthology relations does not seem to appear in the literature, even though Falls et al. [2008] describes clusters of orthologous genes as Turán graphs, a subclass of cographs. Related methods, which use clustering algorithms to help identify orthologs, have been developed e.g. by Tatusov et al. [2000], Li et al. [2003], Berglund et al. [2008], Wheeler et al. [2008] and Lechner et al. [2011].

We suspect that Corollary 26 could have far-reaching consequences for the area of orthology detection. In particular, instead of employing clustering techniques, given an arbitrary orthology relation

δ, it suggests looking for either a symbolic ultrametric or a cograph that is 'close' to δ, from which a (partially resolved) gene tree could then be constructed. Clearly this is not a trivial endeavor since in practical applications any estimate of δ will be plagued by noise and hence will be neither a symbolic ultrametric nor a cograph.

For finding cographs there is a large literature that could be useful for analyzing orthology relations. For example, in the cograph editing problem, given a graph $G = (V,E)$ one aims to convert G into a cograph $G^* = (V,E^*)$ such that the number $|E \triangle E^*|$ of inserted or deleted edges is minimized. Recently it has been proven that this optimization problem is NP-complete [Liu et al., 2011] which, in view of the above results, implies the following:

Corollary 27. *Let $\delta : X \times X \to \{0,1\}^{\odot}$ be an orthology relation map, and K be a positive integer. Then the problem of deciding if there is a map $\delta^* : X \times X \to \{0,1\}^{\odot}$ such that $G_1(\delta^*)$ is a cograph (or, equivalently, δ^* a symbolic ultrametric) with $|E_1(\delta) \triangle E_1(\delta^*)| \leq K$ is NP-complete.*

Even so, it should be noted that the cograph editing problem is fixed parameter tractable [Protti et al., 2009], and so there may be off-the-shelf solutions to help get around this difficulty. Alternatively, efficient Integer Linear Programming (ILP) approaches might be worth investigating.

CHAPTER 6

Partitions, Pseudo-Cherries and Cliques

In this chapter, we will show that the cliques in a certain graph $G(\delta)$ that can be associated to a symbolic ultrametric $\delta : X \times X \to M^{\odot}$ are closely related to the structure of the discriminating symbolic representation of δ. We use this result to help derive a new algorithm for determining whether a map is a symbolic ultrametric or not. We shall also show that cliques in $G(\delta)$ can be characterized in terms of cliques in the graphs $G_m(\delta)$, $m \in M$, defined in the previous chapter.

6.1 From Partitions and Pseudo-Cherries to Cliques

In this section we present a connection between symbolic ultrametrics and a certain collection of partitions that can be associated with the corresponding tree (see Corollary 31). We will later use this result to help obtain a new algorithm for deciding whether or not a map is a symbolic ultrametric and, if this is the case, for constructing its corresponding tree representation.

6.1.1 Partitions and Pseudo-Cherries

Let $\delta : X \times X \to M^{\odot}$ be a symmetric map that satisfies (U0) of Definition 2.2. For $\delta(x,y) = m \in M$, $x \neq y$, we have $\{x,y\} \subseteq N_m(x) \triangle N_m(y)$, where $\triangle$ denotes the usual symmetric difference of sets. For future reference note that, with $N_m[x] := N_m(x) \cup \{x\}$, $x \in X$, we have

$$N_m(x) \triangle N_m(y) = \{x,y\} \text{ if and only if } N_m[x] = N_m[y], \tag{6.1}$$

for all $m \in M$ and all $x,y \in X$. Also note that this condition is satisfied for at most one $m \in M$ for any given pair $\{x,y\} \in \binom{X}{2}$.

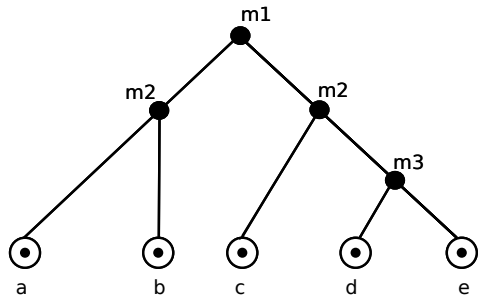


Figure 6.1: A phylogenetic tree T on the set $X = \{a, \ldots, e\}$, together with a map t from the set of interior vertices of T to the set of events $M = \{m_1, m_2, m_3\}$. The corresponding graph $G(\delta)$ is the graph with vertex set $\{a, \ldots, e\}$ and edge set $\{\{a,b\},\{d,e\}\}$.

Now, define $G(\delta)$ to be the graph with vertex set X and edge set

$$E(\delta) := \left\{ \{x,y\} \in \binom{X}{2} : N_m[x] = N_m[y] \text{ for some } m \in M \right\}. \tag{6.2}$$

For example, if $\delta = d_{(T;t)}$ for the pair $(T;t)$ depicted in Fig. 6.1, then the graph $G(\delta)$ is the graph with vertex set $\{a, \ldots, e\}$ and edge set $\{\{c,e\},\{a,d\}\}$.

Suppose that T is a phylogenetic tree on X with root ρ. Let $C \subseteq X$ be a non-empty subset of X and put $v_C = \mathrm{lca}_T(C)$. We call C a *pseudo-cherry* of T if a leaf x of T is adjacent to v_C if and only if $x \in C$. If, in addition, every vertex $v \in V(T)$ adjacent to v_C that does not lie on a path from ρ to v_C is contained in X, then we call C a *cherry* of T. Note that a pseudo-cherry must contain at least one element and that the definition of a cherry C reduces to the usual definition of a cherry (as given e.g. by [Semple and Steel, 2003]) in case $|C| = 2$. We illustrate these two definitions in Fig. 6.2.

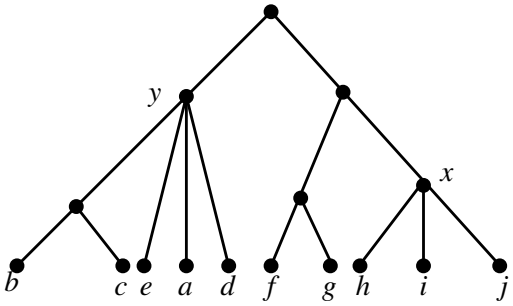


Figure 6.2: A phylogenetic tree T on $X = \{a, b, c, \cdots, j\}$. The vertices $x = \mathrm{lca}_T(C')$ and $y = \mathrm{lca}_T(C)$ are the most recent common ancestors of the sets $C = \{a,d,e\}$ and $C' = \{h,i,j\}$. Both C and C' are pseudo-cherries of T. However, C' is also a cherry of T whereas C is not.

Now, let $t : V(T) \to M^{\odot}$ be a symbolic dating map for T. For each $m \in M$, we define a relation $\sim_m$ on X by putting, for all $x, y \in X$, $x \sim_m y$ if $x = y$ or, in case x and y are distinct, $t(u) = m$ holds for every interior vertex u of T that lies on the unique path from x to y. Clearly $\sim_m$ is an equivalence relation

on X. We write $\widetilde{\Pi}_m$ for the corresponding partition of X. Note that the $\sim_m$-equivalence classes can in some cases be estimated directly from data without having to construct a tree (e.g. for inparalogs, that is, paralogs which all arise from duplication events after a speciation event). Fig 6.3 shows an example of this partition.

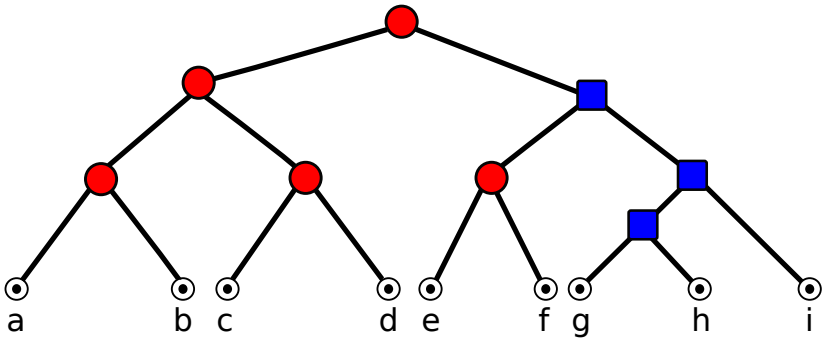


Figure 6.3: An example of the partition $\widetilde{\Pi}_m$. A phylogenetic tree T on the set $X = \{a, ..., i\}$. For every u of T in the unique path from a to c it holds $t(u) = \bullet$, similarly, for every v of T in the unique path from g to i, $t(v) = \blacksquare$ holds.

6.1.2 Cliques

We now show that if δ is a symbolic ultrametric, then the cliques in the graph $G(\delta)$ correspond to pseudo-cherries in the discriminating symbolic representation of δ.

Proposition 28. *Let T be a phylogenetic tree on X, $t : V(T) \to M^{\odot}$ be a symbolic dating map, and $\delta = d_{(T;t)}$ be the associated symbolic ultrametric on X. Then:*

(i) $x \sim_m y$ if and only if $N_m[x] = N_m[y]$, for all $\{x,y\} \in \binom{X}{2}$ and all $m \in M$.

(ii) The graph $G(\delta)$ is the disjoint union of its maximal cliques.

(iii) If the map t is discriminating, then a non-empty subset C of X is a maximal clique of $G(\delta)$ if and only if C is a pseudo-cherry of T.

Proof. (i) Suppose first that $\{x,y\} \in \binom{X}{2}$ such that $x \sim_m y$ for some $m \in M$. Assume for contradiction that $N_m[x] \neq N_m[y]$, that is, $(N_m(x) \triangle N_m(y)) \setminus \{x,y\} \neq \emptyset$, in view of Equ. (6.1). Choose some element z in that set. Then the restriction $T' := T|\{x,y,z\}$ of T to $\{x,y,z\}$ is either the star with leaf set $\{x,y,z\}$ or isomorphic to one of the triples in $\mathscr{R} := \{xy|z, yz|x, xz|y\}$. If T' were the star on $\{x,y,z\}$ then $\mathrm{lca}_T(x,z) = \mathrm{lca}_T(z,y) = \mathrm{lca}_T(x,y)$ would follow. But then $\delta(x,z) = \delta(z,y) = \delta(x,y) = m$ so that $z \in N_m(x) \cap N_m(y)$, contradicting $z \in N_m(x) \triangle N_m(y) - \{x,y\}$. Thus T' must be isomorphic to one of the triples in $\mathscr{R}$.

If T' were isomorphic to the triple $xy|z$ then $\mathrm{lca}_T(x,z) = \mathrm{lca}_T(y,z)$ and so $\delta(x,z) = \delta(y,z)$. But the choice of z implies that we may assume without loss of generality that $z \in N_m(x) \setminus N_m(y)$, so

that $\delta(x,z) = m \neq \delta(y,z)$, a contradiction. If T' were isomorphic to the triple $xz|y$, then $\mathrm{lca}_T(x,y) = \mathrm{lca}_T(z,y)$ and so $\delta(z,y) = \delta(x,y) = m$ would follow as, by assumption, $x \sim_m y$ holds. But then $z \notin N_m(x) \setminus N_m(y)$, a contradiction. By symmetry, T' cannot be isomorphic to the remaining triple $yz|x$ either which yields the final contradiction.

Conversely, suppose that $\{x,y\} \in \binom{X}{2}$ such that $N_m[x] = N_m[y]$, some $m \in M$. We need to show that $t(w) = m$ holds for every interior vertex $w \in V(T)^0$ on the path P from x to y. Assume, for contradiction, that there exists some interior vertex $u \in V(T)^0$ on P with $t(u) \neq m$. Then $u \neq \mathrm{lca}_T(x,y)$ since $t(\mathrm{lca}_T(x,y)) = \delta(x,y) = m$ as, by assumption, $N_m[x] = N_m[y]$. Starting at x and traversing P, let $u' \in V(P)$ and $u'' \in V(P)$ denote the predecessor and successor of u, respectively. Since T is an phylogenetic tree and so has no vertex with in- and outdegree one, there must exist a leaf $z \in V(T)$ such that the path from u to z does not cross the edges $\{u',u\},\{u'',u\} \in E(P)$. Thus $z \notin \{x,y\}$ and either $\mathrm{lca}_T(x,z) = u$ or $\mathrm{lca}_T(y,z) = u$ must hold. By symmetry, we may assume without loss of generality that $\mathrm{lca}_T(x,z) = u$. Then $\delta(x,z) = t(u) \neq m$ and so $z \notin N_m(x)$. By construction of z, we have $\mathrm{lca}_T(y,z) = \mathrm{lca}_T(y,x)$ and so $\delta(y,z) = \delta(y,x) = m$. Hence, $z \in N_m(y)$ and so $z \in N_m(y) \setminus N_m(x) \subseteq N_m(y) \triangle N_m(x) = \{x,y\}$. This is a contradiction in view of Equ. (6.1). Thus, $t(w) = m$ for every interior vertex $w \in V(T)^0$ on P, as required.

(ii) The observation that $G(\delta)$ is the disjoint union of its maximal cliques is a trivial consequence of (i) and the fact that $\sim_m$ is an equivalence relation on X, for all $m \in M$.

(iii) Suppose that t is discriminating. Then the definition of a pseudo-cherry immediately implies that any pseudo-cherry of T must be a maximal clique of $G(\delta)$.

Conversely, assume that C is a maximal clique of $G(\delta)$. Put $v_C := \mathrm{lca}_T(C)$. We show first that every leaf of T adjacent with v_C must be contained in C. To see this, note that if there is a leaf $z \in X - C$ of T adjacent to v_C then $\mathrm{lca}_T(z,x) = v_C$ would hold for all $x \in C$ and, so, $\delta(z,x) = t(v_C)$ would follow for all such x. But then $C \cup \{z\}$ would be a clique in $G(\delta)$ that contains C which is impossible as C is a maximal clique in $G(\delta)$.

Now, for contradiction, assume that there exists some leaf $z \in V(T)$ of T that is contained in C but is not adjacent to v_C. Then, by the definition of v_C, we must have $|C| \geq 2$. Put $m = t(v_C)$ and note that $\delta(x,y) = m$ holds for all $\{x,y\} \in \binom{C}{2}$. Also note that the path P from v_C to z must be of length at least two. Let $w \in V(T)^0$ denote the child of v_C on P. Since t is discriminating, it follows that $t(w) \neq m$. Let $y \in X$ be a leaf of T for which there exists a directed path from w to y and this path does not have an edge in common with the path from w to z. Note that $y \in C$ cannot hold since $\mathrm{lca}_T(z,y) = w$ and, so, $\delta(z,y) = t(w) \neq m$. Thus, $y \in X - C$. Since $y \notin N_m(z)$ and $y \in N_m(x)$ clearly holds for all $x \in C$, we obtain $y \in N_m(x) \Delta N_m(z) = \{x,z\}$ in view of the fact that C is a clique, $x,z \in C$, and Equ. (6.1), a contradiction. Thus, C is a pseudo-cherry of T. □

6.1.3 Maximal Cliques

We denote by $\mathfrak{C}(G)$ the (set-inclusion) maximal cliques of a graph G, and for brevity we let $\mathfrak{C}(\delta)$ denote $\mathfrak{C}(G(\delta))$, for $\delta : X \times X \to M^{\odot}$ a symmetric map that satisfies (U0). Note that $\delta(x,y) = \delta(u,v)$

holds for any clique $C \in \mathfrak{C}(\delta)$ with $|C| \geq 2$ and any two $\{x,y\},\{u,v\} \in \binom{C}{2}$ in case δ is a symbolic ultrametric. Also note that there exists a $C \in \mathfrak{C}(\delta)$ with $|C| \geq 2$ because the tree T_δ has a vertex such that all of its children (of which there must be at least two) are leaves.

We now give an alternative description of the maximal cliques of $G(\delta)$ for δ a symbolic ultrametric, in terms of the graphs $G_m(\delta)$ defined in the last chapter. To this end, we first describe a general way of constructing a partition from a collection of subsets of a non-empty, finite set. Denote the power set of a non-empty, finite set Y by $\mathscr{P}(Y)$ and assume that Z is a finite, non-empty set. We say that a collection $\mathfrak{A} \in \mathscr{P}(Z)$ is a *cover for* Z if $\bigcup_{A\in\mathfrak{A}} A = Z$ holds. Now, suppose $\mathfrak{A} \in \mathscr{P}(Z)$ is a cover for Z. Then we associate to $\mathfrak{A}$ a collection $\Pi(\mathfrak{A})$ of subsets $B \subseteq Z$ that satisfy the following three conditions:

(P1) there exists some $A \in \mathfrak{A}$ such that $B \subseteq A$,

(P2) there are no two distinct elements $x,y \in B$ such that there exists some $A \in \mathfrak{A}$ with $x \in A$ and $y \notin A$, and

(P3) B is (set-inclusion) maximal with respect to satisfying Property (P2).

The proof of the following lemma is routine.

Lemma 29. *Suppose Z is a non-empty, finite set. If a collection $\mathfrak{A}$ of subsets of Z is a cover for Z then $\Pi(\mathfrak{A})$ is a partition of Z.*

Now, suppose $\delta : X \times X \to M^\odot$ is map that satisfies Properties (U0) and (U1). Then, for all $m \in M$, Lemma 29 implies that $\Pi(\mathfrak{C}(G_m(\delta)))$ is a partition of X, since any vertex of a graph must be a vertex in a maximal clique of that graph. For example, consider again the symbolic ultrametric $\delta = d_{(T;t)}$ associated to the pair $(T;t)$ depicted in Fig. 6.1. Then $\Pi(\mathfrak{C}(G_{m_3}(\delta))) = \{\{d,e\}\}$ and $\Pi(\mathfrak{C}(G_{m_2}(\delta))) = \{\{a,b\}\}$ and $\Pi(\mathfrak{C}(G_{m_1}(\delta)))$ is the partition that consist of all singletons of $\{a,\ldots,e\}$.

We now show that for all $m \in M$ the partition $\widetilde{\Pi}_m$ corresponding to the equivalence relation $\sim_m$ defined above can be given in terms of the cliques of $G_m(\delta)$.

Theorem 30. *Let T be a phylogenetic tree on X and let $t : V(T) \to M^\odot$ be a symbolic dating map. Then $\Pi(\mathfrak{C}(G_m(d_{(T;t)}))) = \widetilde{\Pi}_m$ holds for all $m \in M$.*

Proof. Suppose $m \in M$ and put $\delta = d_{(T;t)}$ and $\Pi_m = \Pi(\mathfrak{C}(G_m(\delta))$. Since both Π_m and $\widetilde{\Pi}_m$ are partitions of X it suffices to show that a subset $A \subseteq X$ with $|A| \geq 2$ is an element in Π_m if and only if it is an element in $\widetilde{\Pi}_m$.

Suppose first that $A \in \widetilde{\Pi}_m$. Let $\{x,y\} \in \binom{A}{2}$. Then $x \sim_m y$ and so $t(\mathrm{lca}_T(x,y)) = m$. Thus $\{x,y\} \in E(G_m(\delta))$. Consequently, there must exist a maximal clique $C \in \mathfrak{C}(G_m(\delta))$ such that $x,y \in C$. Without loss of generality, we may assume that C is of minimal size with this property. Since A is a maximal clique in $\mathfrak{C}(\delta)$ it follows that $A \subseteq C$ and that there cannot exist some $C' \in \mathfrak{C}(G_m(\delta))$ and distinct $x,y \in A$ such that $x \in C'$ and $y \notin C'$. But then A satisfies Properties (P1) – (P3) with regards to $\mathfrak{C}(G_m(\delta))$ and so $A \in \Pi_m$ must hold.

Conversely, suppose $A \in \Pi_m$ and assume for contradiction that A is not an equivalence class in $\widetilde{\Pi}_m$. Let $\{x,y\} \in \binom{A}{2}$. Then there must exist some interior vertex v on the path P from x to y in T such that $t(v) \neq m$. Since $A \in \Pi_m$ we cannot have $v = \mathrm{lca}_T(x,y)$. Assume without loss of generality that v lies on the path from $\mathrm{lca}_T(x,y)$ to the leaf x. Also assume without loss of generality that v is such that $t(w) = m$ holds for all interior vertices w on the path P' from v to x. Since T does not have vertices of degree two (except possibly the root of T) there must exist a child w of v that is not a vertex of P'. Let $z \in V(T)$ denote a leaf of T such that w lies on the path from v to z. Since X is the vertex set of $G_m(\delta)$ and $t(\mathrm{lca}_T(z,y)) = m \neq t(v) = t(\mathrm{lca}_T(x,z))$ there must exist some $D \in \mathfrak{C}(G_m(\delta))$ such that $z,y \in D$ and $x \notin D$. But this is impossible as $x,y \in A$ and $A \in \Pi_m$. □

As a consequence we now immediately obtain the aforementioned relationship:

Corollary 31. *Suppose $\delta : X \times X \to M^{\odot}$ is a symbolic ultrametric. Then the maximal cliques of $G(\delta)$ are the set-inclusion maximal subsets in $\bigcup_{m \in M} \Pi(\mathfrak{C}(G_m(\delta)))$.*

Proof. The statement follows from Proposition 28(ii) and (iii), the fact that a non-empty subset C of X is a pseudo-cherry of T_δ if and only if $A \in \widetilde{\Pi}_m$ holds for some $m \in M$, and Theorem 30. □

6.2 A Bottom-Up Construction of Symbolic Representations

We have seen in Proposition 20 that the `BUILD` algorithm can be used to determine whether a map is a symbolic ultrametric or not, and if so, constructs its discriminating symbolic representation. `BUILD` can be thought of as a "top-down" algorithm as, in essence, it starts at the root of the tree (if it exists) and ends when it reaches the leaves. In this section, we present an alternative "bottom-up" algorithm, called `BOTTOM-UP`, which will use our clique-based analysis of symbolic representations in the last section. Such an algorithm could provide a potentially useful alternative to `BUILD` as it is based on finding (nearly) maximal cliques in graphs, for which many different algorithms have been developed in the literature (see e.g. Bron and Kerbosch [1973], Eblen et al. [2011], Kazuhisa and Takeaki [2004], Schmidt et al. [2009]).

Suppose that $\delta : X \times X \to M^{\odot}$ is a symbolic ultrametric, and that $(T;t)$ is some symbolic representation of δ. For every maximal clique $C \in \mathfrak{C}(\delta)$ let $x_C \in C$ denote an arbitrary but fixed element in C. Then it is easy to check that the map

$$d'_{(T,t)} : \mathfrak{C}(\delta) \times \mathfrak{C}(\delta) \to M^{\odot},\ d'_{(T;t)}(C,C') = d_{(T;t)}(x_C, x_{C'}), \tag{6.3}$$

$C, C' \in \mathfrak{C}(\delta)$, is well-defined. A key observation that we shall use in the `BOTTOM-UP` algorithm is that the map $d'_{(T,t)}$ is in fact a symbolic ultrametric on $\mathfrak{C}(\delta)$.

In order to prove this last statement, we shall associate a phylogenetic tree T' on $\mathfrak{C}(\delta)$ plus a symbolic dating map $t' : \mathfrak{C}(\delta) \to M^{\odot}$ for T' as follows. Note that by Proposition 28, every element in $\mathfrak{C}(\delta)$ is a pseudo-cherry of T_δ; we put $v_C = \mathrm{lca}_{T_\delta}(C)$, for all $C \in \mathfrak{C}(\delta)$, and fix some leaf $x_C \in L(T_\delta)$

contained in C. Next, we remove all leaves in $C \setminus \{x_C\}$ from T together with all edges in $\{\{v_C, y\} \in E(T) : y \in C \setminus \{x_C\}\}$. If $v_C \neq \rho_T$ and this process has rendered v_C a vertex of degree two then suppress v_C, and if $v_C = \rho_T$ and this process has rendered v_C a vertex of outdegree one then identify v_C with its unique leaf. Let $T' = (V', E')$ denote the resulting tree. Then the restriction $t'|_{V'}$ of t to V' is clearly a discriminating symbolic dating map for T'. Moreover, since

$$d'_{(T;t)}(C,C') = d_{(T;t)}(x_C, x_{C'}) = t(\mathrm{lca}_T(x_C, x_{C'})) = t'(\mathrm{lca}_{T'}(C,C')) = d_{(T';t')}(C,C')$$

holds for all $C, C' \in \mathfrak{C}(\delta)$, it follows that $(T';t')$ is the (necessarily unique) discriminating symbolic representation of $d'_{(T;t)}$. Thus, by Theorem 15 we have:

Proposition 32. *Let T be a phylogenetic tree on X and $t : V(T) \to M^{\odot}$ be a symbolic dating map. Then the map $d'_{(T;t)} : \mathfrak{C}(d_{(T;t)}) \times \mathfrak{C}(d_{(T;t)}) \to M^{\odot}$ defined in Equ. 6.3 is a symbolic ultrametric on $\mathfrak{C}(d_{(T;t)})$.*

We now establish a second result which will be central to the `BOTTOM-UP` algorithm. Given a map $\delta : X \times X \to M^{\odot}$ satisfying (U0)–(U2) we denote the set of connected components of $G(\delta)$ by $\pi(\delta)$ and, for future reference, we let $\pi_2(\delta)$ denote those elements in $\pi(\delta)$ with size at least two.

Lemma 33. *Suppose that $\delta : X \times X \to M^{\odot}$ is a map that satisfies Properties (U0)–(U2), and $K \in \pi_2(\delta)$. Then the following hold:*

(i) If $\{x,y,z\} \in \binom{K}{3}$ is such that x,y,z is a path in K of length two, then $\delta(x,y) = \delta(y,z)$.

(ii) If $\{x,y,z\} \in \binom{K}{3}$ is such that $\{x,y\}$ and $\{y,z\}$ are edges in K, then $\{z,x\}$ must also be an edge in K.

(iii) K is a clique and $\delta(x,y) = \delta(u,v)$ holds for all $\{x,y\}, \{u,v\} \in \binom{K}{2}$.

Proof. (i) Suppose for contradiction that there exists $\{x,y,z\} \in \binom{K}{3}$ such that x,y,z is a path of length two but $m_1 := \delta(x,y) \neq \delta(y,z) =: m_2$. Then Property (U2) implies $\delta(x,z) \in \{m_1, m_2\}$. Without loss of generality we may assume that $\delta(x,z) = m_1$. Then $z \in N_{m_1}(x)$ and, since $\delta(x,z) \neq m_1$, we also have $z \notin N_{m_1}(y)$. Hence, $z \in N_{m_1}(x) - N_{m_1}(y) \subseteq N_{m_1}(x) \Delta N_{m_1}(y) = \{x,y\}$ since $\{x,y\}$ is an edge in K, a contradiction.

(ii) Suppose for contradiction that there exists $\{x,y,z\} \in \binom{K}{3}$ such that $\{x,y\}$ and $\{y,z\}$ are edges in K but $\{x,z\}$ is not an edge in K. Then Assertion (i) implies that $\delta(x,y) = \delta(y,z) =: m$. We distinguish the cases $\delta(x,z) = m$ and $\delta(x,z) \neq m$.

First, suppose $\delta(x,z) = m$. Then there must exist some $u \in K - \{x,z\}$ such that $u \in N_m(x) \Delta N_m(z)$ as otherwise $\{x,z\}$ would be an edge in K. Without loss of generality, we may assume that $u \in N_m(x) - N_m(z)$. Note that since both $\{x,y\}$ and $\{y,z\}$ are edges in K it follows that $y \in N_m(x) \cap N_m(z)$ and so $u \neq y$. Moreover, since $\{x,y\}$ is an edge in K, $\{x,y\} = N_m(x) \Delta N_m(y)$ must hold, and so $u \in N_m(y)$. Similarly, since $\{y,z\}$ is an edge in K, $u \in N_m(z)$ which is impossible. Thus $\{x,z\}$ must be an edge of K.

Now suppose $\delta(x,z) \neq m$. Then $z \notin N_m(x)$. Since $\{y,z\}$ is an edge in K we have $z \in N_m(y)$ and so $z \in N_m(y) - N_m(x) \subseteq N_m(y)\Delta N_m(x) = \{x,y\}$ as $\{x,y\}$ is an edge of K, a contradiction. Thus $\{x,z\}$ must be also an edge of K in this case.

(iii) This is an immediate consequence of Assertions (i) and (ii). □

We now present the BOTTOM-UP algorithm. The pseudo-code for this algorithm is given in Algorithm 2. BOTTOM-UP works in a similar way to the UPGMA algorithm [Sneath and Sokal, 1973] for constructing phylogenetic trees from distance matrices. Essentially BOTTOM-UP works by iteratively looking for pseudo-cherries and, if it finds them, defining a new map on the set of these pseudo-cherries along the lines of Proposition 32.

We now prove a result that is analogous to Proposition 20.

Theorem 34. *Suppose $\delta : X \times X \to M^{\odot}$ is a map. Then the algorithm BOTTOM-UP is a polynomial-time algorithm that either:*

(i) outputs a symbolic discriminating representation for δ if δ is a symbolic ultrametric, or

(ii) the statement "δ is not a symbolic ultrametric"

Proof. We first remark that if the input map $\delta : X \times X \to M^{\odot}$ satisfies Properties (U0)–(U2) then, at each execution step of the while loop at Line 3, if Line 5 is not executed then the map δ' defined in Line 12 must also satisfy (U0)–(U2). Moreover, the map t_C defined in Line 8 is well-defined since, in view of Lemma 33, $\delta(C_1,C_2) = \delta(C_3,C_4)$ holds for all $\{C_1,C_2\},\{C_3,C_4\} \in \binom{C}{2}$. In addition, since the set of connected components of a graph can be found in polynomial time and the size of the set F defined in Line 11 decreases by at least one in each execution of the while loop in Line 7 (in case Line 9 is not executed), it follows that the run time for BOTTOM-UP is polynomial in $|X|$.

Now, to complete the proof, given a map $\delta : X \times X \to M^{\odot}$ we will prove the following claims: (i) if δ is a symbolic ultrametric, then BOTTOM-UP will output a phylogenetic tree T on X and a discriminating symbolic dating map for T, and (ii) if BOTTOM-UP returns a phylogenetic tree T on X and a discriminating symbolic dating map t on T, then $(T;t)$ is a discriminating symbolic representation for δ. This will complete the proof of the theorem in view of Theorem 15.

Proof of (i): Assume δ is a symbolic ultrametric so that, in particular, δ satisfies Properties (U0)–(U2). We first remark that, since $\pi_2(\delta) \neq \emptyset$ (in view of Proposition 28(iii)), Line 11 is not executed at the first execution of the while loop on Line 15. Moreover, as in each execution step of that loop the map δ' defined in Line 15 is a symbolic ultrametric, in view of Proposition 32 we must also have $\pi_2(\delta') \neq \emptyset$.

We now show that BOTTOM-UP returns a pair $(T^{\delta};t^{\delta})$ where T^{δ} is a phylogenetic tree on X and t^{δ} is a discriminating symbolic dating map for T^{δ}. Note that it suffices to show that at the end of each execution of the while loop in Line 7, every element $(T_C;t_C)$ in the set F defined at Line 14 consists of a phylogenetic tree T_C and a discriminating symbolic dating map t_C for T_C.

Algorithm 2 BOTTOM-UP

Input: Non-empty finite sets X and M with $|X| \geq 3$ and a map $\delta : X \times X \to M^{\odot}$.

Output: Discriminating symbolic representation of δ or the statement "δ is not a symbolic ultrametric on X".

1 **if** δ *does not satisfy Property (U0), (U1), or (U2)* **then**
2 return the statement "δ is not a symbolic ultrametric on X" and stop.
3 **end**
4 **forall the** $x \in X$ **do**
5 $F = \{(T_{\{x\}}, t_{\{x\}})\}$, where $T_{\{x\}}$ is the tree consisting of one vertex x and $t_{\{x\}}$ is the map on $V(T_{\{x\}})$ given by putting $t_{\{x\}}(x) = \odot$.
6 **end**
7 **while** $|F| \geq 2$ **do**
8 Compute the sets $\pi(\delta)$ and $\pi_2(\delta)$.
if $\pi_2(\delta) = \emptyset$ **then**
9 return the statement "δ is not a symbolic ultrametric on X" and stop.
10 **end**
11 **forall the** $C \in \pi_2(\delta)$ **do**
12 Let T_C be the phylogenetic tree obtained by adding a new vertex w and edges $\{w, \rho_{C'}\}$ from w to the root $\rho_{C'}$ of $T_{C'}$, for all $C' \in C$.
Define $t_C : V(T_C) \to M^{\odot}$ by putting $t_C(w)$ equal to $\delta(C_1, C_2)$ for any $C_1 \neq C_2 \in C$, and $t_{C'}(v)$ for any $v \in V(T_{C'})$, $C' \in C$.
Collapse edges of T_C as necessary to ensure that the restriction of t_C to the vertex set of the resulting tree is discriminating. Denote the resulting pair also by $(T_C; t_C)$.
13 **end**
14 Let $F = \{(T_C; t_C) : C \in \pi(\delta)\}$, where we identify each singleton set in $\pi(\delta)$ with its unique element.
forall the $C \in \pi(\delta)$ **do**
15 choose some $x_C \in C$ and define $\delta' : \pi(\delta) \times \pi(\delta) \to M^{\odot}$ to be the map given by setting $\delta'(C_1, C_2) := \delta(x_{C_1}, x_{C_2})$ for all $C_1 \neq C_2 \in \pi(\delta)$.
16 **end**
17 Let $\delta = \delta'$.
18 **end**
19 Return the unique element in F.

To this end, assume that $k \geq 1$ executions of that loop have been carried out, and denote the map computed in Line 15 at execution l by δ_l, for $l = k-1, k$, where we set $\delta_0 := \delta$. Let $C \in \pi(\delta_{k-1})$. If $C \notin \pi_2(\delta_{k-1})$ then, by assumption, T_C and t_C are of the required form, where we identify C with

its unique element. So assume that $C \in \pi_2(\delta_{k-1})$. Then, by construction, the tree T_C generated in Line 12 is a phylogenetic tree on $\bigcup_{A\in C} L(T_A)$. Since δ satisfies properties (U0)–(U2), the map t_C defined is well-defined in view of the remark at the beginning of the proof. Now, note that there can be at most one $C' \in C$ such that $t_C(w) = t_C(\rho_{C'})$. If there exists no such element, then t_C is a discriminating symbolic dating map for T_C. Moreover, if such an element C' exists, then the map obtained by restricting t_C to the vertex set of the phylogenetic tree obtained from T_C by collapsing the edge $\{w, \rho_{C'}\}$ is a discriminating symbolic dating map for that tree. Thus the pair $(T_C; t_C)$ is of the required form and so (i) follows.

Proof of (ii): Suppose that δ is an arbitrary map, and that BOTTOM-UP returns a pair $(T^\delta; t^\delta)$ with T^δ a phylogenetic tree on X and t^δ a discriminating symbolic dating map for T^δ. Note that in this case, δ must satisfy Properties (U0)–(U2). To show that (ii) holds, it suffices to show that in each execution of the while loop in Line 7 every element $(T_C; t_C)$ in the set F defined in Line 14 is a discriminating symbolic representation of δ restricted to $L(T_C)$.

To this end, assume that $k \geq 1$ executions of the while loop have been carried out and, as before, denote the map defined in Line 15 at execution l by δ_l, $l = k-1, k$ where $\delta_0 := \delta$. Let $C \in \pi(\delta_{k-1})$. If $C \notin \pi_2(\delta_{k-1})$ then, by assumption, $(T_C; t_C)$ is a discriminating symbolic representation for δ restricted to $L(T_C)$, where we identify C with its unique element.

So, assume that $C \in \pi_2(\delta_{k-1})$. Suppose $x, y \in L(T_C)$. Since, by assumption, $(T_{C'}, t_{C'})$ is a discriminating symbolic representation of δ restricted to $L(T_{C'})$, for all $C' \in C$, we may assume without loss of generality that there exist distinct $C_1, C_2 \in C$ such that $x \in L(T_{C_1})$ and $y \in L(T_{C_2})$. Note that the definition of the tree T_C and the map t_C imply that $w = \mathrm{lca}_{T_C}(c_1, c_2)$ holds for all $c_1 \in L(T_{C_1})$ and all $c_2 \in L(T_{C_2})$, and so $\delta(c_1, c_2) = t_C(w)$ for all such c_1 and c_2. But then $d_{(T_C;t_C)}(x, y) = t_C(\mathrm{lca}_{T_C}(x, y)) = t_C(w) = \delta(x, y)$. Thus, again, $(T_C; t_C)$ is a discriminating symbolic representation of δ restricted to $L(T_C)$. This completes the proof of (ii). □

6.3 Concluding Remarks

With the new characterizations for symbolic ultrametrics, in terms of partitions, pseudo-cherries and cliques we have presented new algorithms for recovering the associated trees, with an emphasis on how these algorithms could be potentially extended to deal with arbitrary orthology relations.

More specifically, for finding symbolic representations, it could be of interest to try modifying the BOTTOM-UP algorithm to enable it to handle arbitrary orthology relations. For example, ideas behind the MIN-CUT supertree algorithm [Semple and Steel, 2000], an algorithm extending BUILD which outputs a tree given *any* set of rooted triples, could be explored, as well as related approaches for finding compatible sets of triples that have as many triples as possible in common with a given set of triples, such as those in e.g. [Byrka et al., 2010a]. Alternatively, Proposition 28 suggests that heuristics for finding maximum cliques (or subsets that are close to being maximum cliques) in graphs might be useful for modifying the BOTTOM-UP algorithm. In this direction it would be as well interesting to

explore graphs whose edges have assigned weights and then apply a modified version of the `MIN-CUT supertree` algorithm for weighted graphs.

CHAPTER 7

From Orthology Relations to Species Tree Inference

The reconstruction of the evolutionary history of a gene family is necessarily based on at least three interrelated types of information. The true phylogeny of the investigated species is required as a scaffold with which the associated gene tree must be reconcilable. Orthology or paralogy of genes found in different species determines whether an internal vertex in the gene tree corresponds to a duplication or a speciation event. Speciation events, in turn, are reflected in the species tree.

According to Fitch's definition [Fitch, 2000], two genes are (co-)orthologous if their last common ancestor in the gene tree represents a speciation event. Otherwise, i.e., when their last common ancestor is a duplication event, they are paralogs. The orthology relation on a set of genes is therefore determined by the gene tree T and an "event labeling" that identifies each interior vertex of T as either a duplication or a speciation event.

In the previous chapter we have shown that a relation on a set of genes is an orthology relation (i.e., it derives from some event-labeled gene tree) if and only if it is a cograph. The orthology relation thus places strong and easily interpretable constraints on the gene tree.

This observation suggests that a viable approach to reconstructing histories of large gene families may start from an empirically determined orthology relation, which can be directly adjusted to conform to the requirement of being a cograph. The result is then equivalent to an (usually incompletely resolved) event-labeled gene tree, which might be refined or used as constraint in the inference of a fully resolved gene tree. In this work we are concerned with the next conceptual step: the derivation of a species tree from an event-labeled gene tree. As we shall see below, this problem is much simpler than the full tree reconciliation problem [Hernandez-Rosales et al., 2012]. Technically, we will approach this problem by reducing the reconciliation map from gene tree to species tree to rooted triples of genes residing in three distinct species. This is related to an approach that was developed in

[Chauve and El-Mabrouk, 2009] for addressing the full tree reconciliation problem.

7.1 Reconciliation Map

A gene tree T arises through a series of events along a species tree S. We consider both T and S as phylogenetic trees with leaf sets L (the set of genes) and B (the set of species), respectively. We assume that $|L| \geq 3$ and $|B| \geq 1$. We consider only gene duplications and gene losses, which take place between speciation events, i.e., along the edges of S. Speciation events are modeled by transmitting the gene content of an ancestral lineage to each of its daughter lineages.

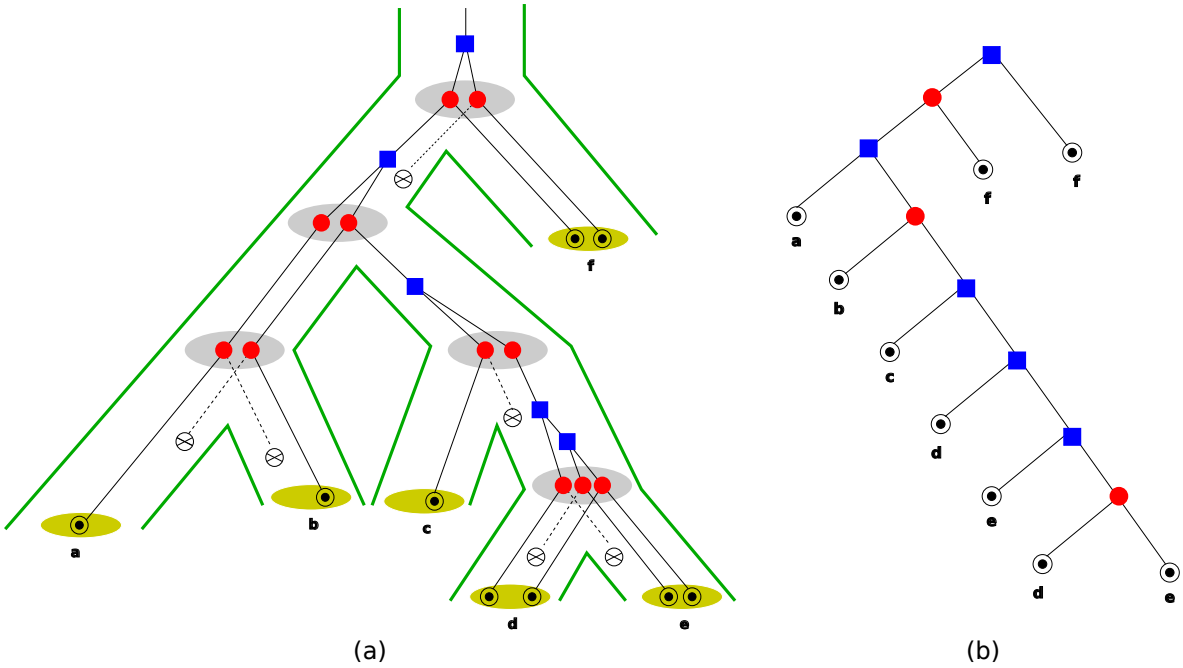


Figure 7.1: **(a)** Example of an evolutionary scenario showing the evolution of a gene family. The corresponding true gene tree $\hat{T}$ appears embedded in the true species tree $\hat{S}$. The map $\hat{\mu}$ is implicitly given by drawing the species tree superimposed on the gene tree. In particular, the speciation vertices in the gene tree (red circles) are mapped to the vertices of the species tree (gray ovals) and the duplication vertices (blue squares) to the edges of the species tree. Gene losses are represented with "$\otimes$" (mapping to edges in $\hat{S}$). The observable species $a, b, \ldots, f$ are the leaves of the species tree (yellow ovals) and extand genes therein are labeled with "$\odot$". **(b)** The corresponding gene tree T with observed events from the tree in (a). Leaves are labeled with the corresponding species.

The true evolutionary history of a single ancestral gene thus can be thought of as a scenario such as the one depicted in Fig. 7.1. Since we do not consider horizontal gene transfer or lineage sorting in this contribution, an evolutionary scenario consists of four components:

1. A true gene tree $\hat{T}$

2. A true species tree $\hat{S}$
3. An assignment of an event type (i.e., speciation ●, duplication ■, loss ⊗, or observable (extant) gene ⊙) to each interior vertex and leaf of $\hat{T}$
4. A map μ assigning every vertex of $\hat{T}$ to a vertex or edge of $\hat{S}$ in such a way that
   (a) the ancestor order of $\hat{T}$ is preserved
   (b) a vertex of $\hat{T}$ is mapped to an interior vertex of $\hat{S}$ if and only if it is of type speciation
   (c) extant genes of $\hat{T}$ are mapped to leaves of $\hat{S}$.

It will be convenient for our discussion below to extend the ancestor relation $\preceq_T$ on V to the union of the edge and vertex sets of T. More precisely, for the directed edge $e = [u,v] \in E$ we put $x \prec_T e$ if and onfly if $x \preceq_T v$ and $e \prec_T x$ if and only if $u \preceq_E x$. For edges $e = [u,v]$ and $f = [a,b]$ in T we put $e \preceq f$ if and only if $v \preceq b$.

In order to allow $\hat{\mu}$ to map duplication vertices to a time point before the last common ancestor of all species in $\hat{S}$, we need to extend our definition of a species tree by adding an extra vertex and an extra edge "above" the last common ancestor of all species. Note that strictly speaking $\hat{S}$ is not a phylogenetic tree anymore. In case there is no danger of confusion, we will from now on refer to a phylogenetic tree on B with this extra edge and vertex added as a species tree on B and to ρ_B as the root of B. Also, we canonically extend our notions of a triple, displaying, etc. to this new type of species tree.

The true gene tree $\hat{T}$ represents all extant as well as all extinct genes, all duplication, and all speciation events. Not all of these events are observable from extant genes data, however. In particular, extinct genes cannot be observed. The observable part $T = T(V,E)$ of $\hat{T}$ is the restriction of $\hat{T}$ to the leaf set L of extant genes, i.e., $T = \hat{T}|_L$.

Furthermore, we can observe a map $\sigma : L \to B$ that assigns to each extant gene the species in which it resides. Of course, for $x \in L$ we have $\sigma(x) = \hat{\mu}(x)$. Here B is the leaf set of the extant species tree, i.e., $B = \sigma(L)$. For ease of readability, we also put $\sigma(T') = \{\sigma(x) : x \in L(y)\}$ for any subtree T' of T with $T' = T(y)$ where $y \in V^0$. Alternatively, we will sometimes also write $\sigma(y)$ instead of $\sigma(T(y))$. Last but not least, for $Y \subseteq L$, we put $\sigma(Y) = \{\sigma(y) : y \in Y\}$.

The observable part of the species tree $S = (W,H)$ is the restriction $\hat{S}|_B$ of $\hat{S}$ to B. In order to account for duplication events that occurred before the first speciation event, the additional vertex $\rho_S \in W$ and the additional edge $[\rho_S, \mathrm{lca}_S B] \in H$ must be part of S.

The evolutionary scenario also implies an *event labeling* map $t : V \to \{●, ■, ⊙\}$ that assigns to each interior vertex v of T a value $t(v)$ indicating whether v is a speciation event (●) or a duplication event (■). It is convenient to use the special label ⊙ for the leaves x of T. We write (T,t) for the event-labeled tree. Here t is a "symbolic dating map" as we introduced it in the previous chapter. It is called *discriminating* if, for all edges $\{u,v\} \in E$, we have $t(u) \neq t(v)$ in which case (T,t) is known to be in 1-1-correspondence to a cograph [Hellmuth et al., 2013]. Note that we will in general not

require that t is discriminating in this contribution. For $T = (V,E)$ a gene tree on L, B a set of species, and maps t and σ as specified above, we require however that μ and σ must satisfy the following compatibility property:

(C) Let $z \in V$ be a speciation vertex, i.e., $t(z) = \bullet$, and let T' and T'' be subtrees of T rooted in two distinct children of z. Then $\sigma(T') \cap \sigma(T'') = \emptyset$.

Note the we do not require the converse, i.e., from the disjointness of the species sets $\sigma(T')$ and $\sigma(T'')$ we do **not** conclude that their last common ancestor is a speciation vertex.

For $x,y \in L$ and $z = \mathrm{lca}_T(x,y)$ it immediately follows from condition (C) that if $t(\mathrm{lca}_T(x,y)) = \bullet$ then $\sigma(x) \neq \sigma(y)$ since, by assumption, x and y are leaves in distinct subtrees below z. Equivalently, two distinct genes $x \neq y$ in L for which $\sigma(x) = \sigma(y)$ holds, that is, they are contained in the same species of B, must have originated from a duplication event, i.e., $t(\mathrm{lca}_T(x,y)) = \blacksquare$. Thus we can regard σ as a proper vertex coloring of the cograph corresponding to (T,t).

Let us now consider the properties of the restriction of $\hat{\mu}$ to the observable parts T of $\hat{T}$ and S of $\hat{S}$. Consider a speciation vertex x in $\hat{T}$. If x has two children y' and y'' so that $L(y')$ and $L(y'')$ are both non-empty then $x = \mathrm{lca}_{\hat{T}}(z',z'')$ for all $z' \in L(y')$ and $z'' \in L(y'')$ and hence, $x = \mathrm{lca}_T(L(y') \cup L(y''))$. In particular, x is an observable vertex in T. Furthermore, we know that $\sigma(L(y')) \cap \sigma(L(y'')) = \emptyset$, and therefore, $\hat{\mu}(x) = \mathrm{lca}_S(\sigma(L(y') \cup L(y''))$. Considering all pairs of children with this property this can be rephrased as $\hat{\mu}(x) = \mathrm{lca}_{\hat{S}}(\sigma(L(x)))$. On the other hand, if x does not have at least two children with this property, and hence the corresponding speciation vertex cannot be viewed as most recent common ancestor of the set of its descendants in S, then x is not a vertex in the restriction $T = \hat{T}|_L$ of $\hat{T}$ to the set L of the extant genes. The restriction μ of $\hat{\mu}$ to the observable tree T therefore satisfies the properties used below to define reconciliation maps.

Definition 35. *Suppose that B is a set of species, that $S = (W,H)$ is a phylogenetic tree on B, that $T = (V,E)$ is a gene tree with leaf set L and that $\sigma : L \to B$ and $t : V \to \{\bullet, \blacksquare, \odot\}$ are the maps described above. Then we say that S is a* species tree for (T,t,σ) *if there is a map $\mu : V \to W \cup H$ such that, for all $x \in V$:*

(i) If $t(x) = \odot$ then $\mu(x) = \sigma(x)$.

(ii) If $t(x) = \bullet$ then $\mu(x) \in W \setminus B$.

(iii) If $t(x) = \blacksquare$ then $\mu(x) \in H$.

(iv) Let $x,y \in V$ with $x \prec_T y$. We distinguish two cases:

1. *If $t(x) = t(y) = \blacksquare$ then $\mu(x) \preceq_S \mu(y)$ in S.*
2. *If $t(x) = t(y) = \bullet$ or $t(x) \neq t(y)$ then $\mu(x) \prec_S \mu(y)$ in S.*

(v) If $t(x) = \bullet$ then $\mu(x) = \mathrm{lca}_S(\sigma(L(x)))$

We call μ the reconciliation map from (T,t,σ) to S.

We note that $\mu^{-1}(\rho_S) = \emptyset$ holds as an immediate consequence of property *(v)*, which implies that no speciation node can be mapped above $\mathrm{lca}_S(B)$, the unique child of ρ_S.

We illustrate this definition by means of an example in Fig. 7.2 and remark that it is consistent with the definition of reconciliation maps for the case when the event labeling t on T is not known. Doyon et al. [2009] presented the following definition for this case:

Definition 36. *Let T be a binary tree with vertices $V(T)$ and edges $E(T)$, such that only its leaves are labeled. Let $r(T), L(T)$ and $\Lambda(T)$ respectively denote its root, the set of its leaves, and the set of the labels of its leaves. Let $\sigma : L(G) \to L(S)$ be the function that maps each leaf of G to the unique leaf of S with the same label. The $LCA-mapping$ $M : V(G) \to V(S)$ maps each vertex u of G to the unique vertex $M(u)$ of S such that $\Lambda(S_{M(u)})$ is the smallest cluster of S containing $\Lambda(G_u)$. A reconciliation between a gene tree G and a species tree S is a mapping $\alpha : V(G) \to V(S) \cup E(S)$ such that:*

1. *For all $u \in L(G), \alpha(u) = M(u) = \sigma(U)$.*
2. *For any vertex $u \in V(G) \setminus L(G)$,*
   - *if $\alpha(u) \in V(S)$, then $\alpha(u) = M(u)$.*
   - *if $\alpha(u) \in E(S)$, then $M(u) \prec_S \alpha(u)$.*
3. *For any two vertices $u, v \in V(G)$, such that $v \prec_G u$,*
   - *if $\alpha(u), \alpha(v) \in E(S)$, then $\alpha(v) \preceq_S \alpha(u)$,*
   - *otherwise, $\alpha(v) \prec_S \alpha(u)$.*

Continuing with our notation from Definition 35 for the remainder of this section, we easily derive their axiom set as

Lemma 37. *If μ is a reconciliation map from (T,t,σ) to S and L is the leaf set of T then, for all $x \in V$,*

(D1) $x \in L$ *implies* $\mu(x) = \sigma(x)$.

(D2.a) $\mu(x) \in W$ *implies* $\mu(x) = \mathrm{lca}_S(\sigma(L(x)))$.

(D2.b) $\mu(x) \in H$ *implies* $\mathrm{lca}_S(\sigma(L(x))) \prec_S \mu(x)$.

(D3) Suppose $x, y \in V$ such that $x \prec_T y$. If $\mu(x), \mu(y) \in H$ then $\mu(x) \preceq_S \mu(y)$; otherwise $\mu(x) \prec_S \mu(y)$.

Proof. Suppose $x \in V$. Then (D1) is equivalent to *(i)* and the fact that $t(x) = \odot$ if and only if $x \in L$. Conditions *(ii)* and *(v)* together imply (D2.a). If $\mu(x) \in H$ then x is duplication vertex of T. From condition *(iv)* we conclude that $\mathrm{lca}_S(\sigma(L(x))) \preceq_S \mu(x)$. Since $\mathrm{lca}_S(\sigma(L(x))) \in W$, equality cannot hold and so (D2.b) follows. (D3) is an immediate consequence of *(iv)*. □

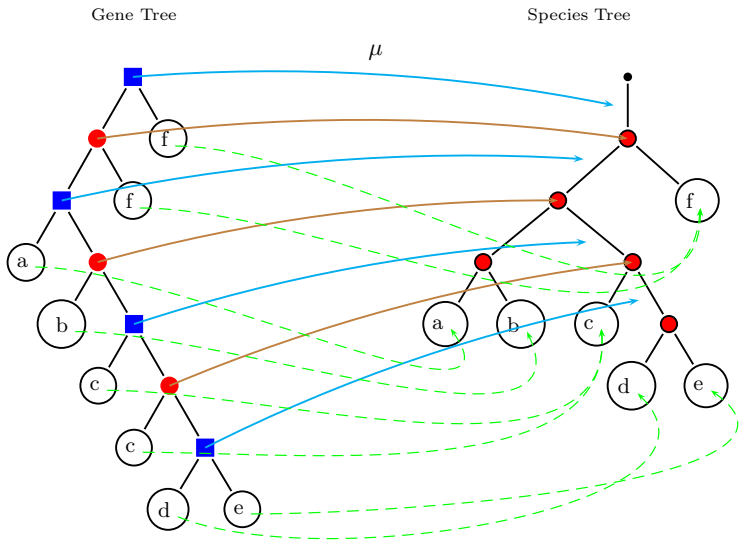


Figure 7.2: Example of the mapping μ of nodes of the gene tree T to the species tree S. Speciation nodes in the gene tree (red circles) are mapped to nodes in the species tree, duplication nodes (blue squares) are mapped to edges in the species tree. σ is shown as dashed green arrows. For clarity of exposition, we have identified the leaves of the gene tree on the left with the species they reside in via the map σ.

For T a gene tree, B a set of species and maps σ and t as above, our goal is now to characterize:

1. Those (T,t,σ) for which a species tree on B exists.
2. Species trees on B that are species trees for (T,t,σ).

Unless stated otherwise, we continue with our assumptions on B, (T,t,σ), and S as stated in Definition 35. We start with the simple observation that a reconciliation map from (T,t,σ) to S preserves the ancestor order of T and hence T imposes a strong constraint on the relationship of most recent common ancestors in S:

Lemma 38. *Let $\mu : V \to W \cup H$ be a reconciliation map from (T,t,σ) to S. Then*

$$\mathrm{lca}_S(\mu(x),\mu(y)) \preceq_S \mu(\mathrm{lca}_T(x,y)) \tag{7.1}$$

holds for all $x,y \in V$.

Proof. Assume that x and y are distinct vertices of T. Consider the unique path P connecting x with y. P is uniquely subdivided into a path P' from x to $\mathrm{lca}_T(x,y)$ and a path P'' from $\mathrm{lca}_T(x,y)$ to y. Condition (iv) implies that the images of the vertices of P' and P'' under μ, resp., are ordered in S with regards to $\preceq_S$ and hence are contained in the intervals Q' and Q'' that connect $\mu(\mathrm{lca}_T(x,y))$ with $\mu(x)$ and $\mu(y)$, respectively. In particular $\mu(\mathrm{lca}_T(x,y))$ is the largest element (w.r.t. $\preceq_S$) in the union of $Q' \cup Q''$ which contains the unique path from $\mu(x)$ to $\mu(y)$ and hence also $\mathrm{lca}_S(\mu(x),\mu(y))$. □

7.2 Inferring species trees from triple sets

Since a phylogenetic tree (in the original sense) T is uniquely determined by its induced triple set $\mathscr{R}_T$, it is reasonable to expect that all the information on the species tree(s) for (T,t,σ) is contained in the images of the triples in $\mathscr{R}_T$ (or more precisely their leaves) under σ. However, this is not the case in general as the situation is complicated by the fact that not all triples in $\mathscr{R}_T$ are informative about a species tree that displays T. The reason is that duplications may generate distinct paralogs long before the divergence of the species in which they eventually appear. To address this problem, we associate to (T,t,σ) the set of triples

$$\mathfrak{G} = \mathfrak{G}(T,t,\sigma) = \left\{ r \in \mathscr{R}_T \middle| t(\mathrm{lca}_T(r)) = \bullet \text{ and } \sigma(x) \neq \sigma(y), \text{for all} x,y \in L(r) \text{pairwise distinct} \right\}. \tag{7.2}$$

As we shall see below, $\mathfrak{G}(T,t,\sigma)$ contains all the information on a species tree for (T,t,σ) that can be gleaned from (T,t,σ).

Lemma 39. *If μ is a reconciliation map from (T,t,σ) to S and $((x,y),z) \in \mathfrak{G}(T,t,\sigma)$ then S displays $((\sigma(x),\sigma(y)),\sigma(z))$.*

Proof. Put $\mathfrak{G} = \mathfrak{G}(T,t,\sigma)$ and recall that L denotes the leaf set of T. Let $\{x,y,z\} \in \binom{L}{3}$ and assume w.l.o.g. that $((x,y),z) \in \mathfrak{G}$. First consider the case that $t(\mathrm{lca}_T(x,y)) = \bullet$. From condition *(v)* we conclude that $\mu(\mathrm{lca}_T(x,y)) = \mathrm{lca}_S(\sigma(x),\sigma(y))$ and $\mu(\mathrm{lca}_T(x,y,z)) = \mathrm{lca}_S(\sigma(x),\sigma(y),\sigma(z))$. Since, by assumption, $\mathrm{lca}_T(x,y) \prec \mathrm{lca}_T(x,y,z)$, we have as a consequence of condition *(iv)* that $\mu(\mathrm{lca}_T(x,y)) \prec \mu(\mathrm{lca}_T(x,y,z))$. From $\mathrm{lca}_T(x,z) = \mathrm{lca}_T(y,z) = \mathrm{lca}_T(x,y,z)$ we conclude that S must display $((\sigma(x),\sigma(y)),\sigma(z))$ as S is assumed to be a species tree for (T,t,σ).

Now suppose that $t(\mathrm{lca}_T(x,y)) = \blacksquare$ and therefore, $\mu(\mathrm{lca}_T(x,y)) \in H$. Moreover, $\mu(\mathrm{lca}_T(x,y,z)) \in W$ holds. Hence, Lemma 38 and property *(iv)* together imply that $\mathrm{lca}_S(\sigma(x),\sigma(y)) \prec_S \mu(\mathrm{lca}_T(x,y)) \prec_S \mu(\mathrm{lca}_T(x,y,z))$. Thus, we again obtain that the triple $((\sigma(x),\sigma(y)),\sigma(z))$ is displayed by S. □

It is important to note that a similar argument cannot be made for triples in $\mathscr{R}_T$ rooted in a duplication vertex of T as such triplets are in general not displayed by a species tree for (T,t,σ). We present the generic counterexample in Fig. 7.3.

To state our main result (Theorem 41), we require a further definition.

Definition 40. *For (T,t,σ), we define the set*

$$\mathfrak{S} = \mathfrak{S}(T,t,\sigma) = \{((a,b),c) | \exists((x,y),z) \in \mathfrak{G}(T,t,\sigma) \text{ with } \sigma(x) = a,\ \sigma(y) = b, \text{ and } \sigma(z) = c\} \tag{7.3}$$

As an immediate consequence of Lemma 39, $\mathfrak{S}(T,t,\sigma)$ must be displayed by any species tree for (T,t,σ) with leaf set B.

Theorem 41. *Let S be a species tree with leaf set B. Then there exists a reconciliation map μ from (T,t,σ) to S whenever S displays all triples in $\mathfrak{S}(T,t,\sigma)$.*

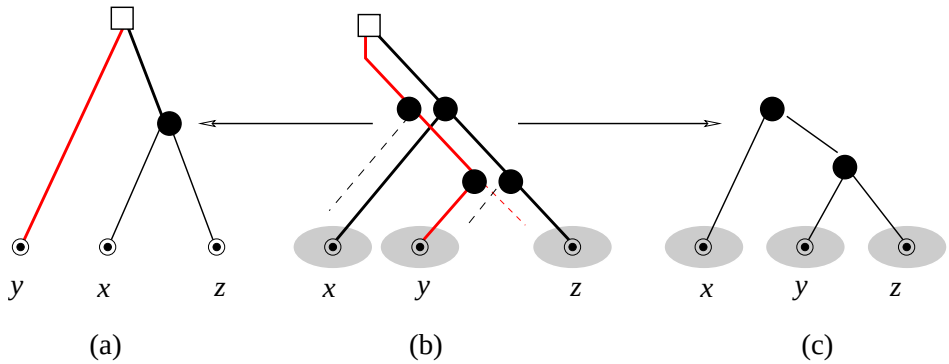


Figure 7.3: Triples from T whose root is a duplication event are in general not displayed from the species tree S. **(a)** Triple with duplication event at the root obtained from the true evolutionary history of T shown in panel **(b)**. Panel **(c)** is the true species tree. In the triple **(a)** the species y appears as the outgroup even though the x is the outgroup in the true species tree.

Proof. Recall that L is the leaf set of $T = (V,E)$. Put $S = (W,H)$ and $\mathfrak{S} = \mathfrak{S}(T,t,\sigma)$. We first consider the subset $G := \{x \in V \mid t(x) \in \{\bullet, \odot\}\}$ of V comprising of the leaves and speciation vertices of T.

We explicitly construct the map $\mu : G \to W$ as follows. For all $x \in V$, we put

(M1) $\mu(x) = \sigma(x)$ if $t(x) = \odot$,

(M2) $\mu(x) = \mathrm{lca}_S(\sigma(L(x)))$ if $t(x) = \bullet$.

Note that alternative (M1) ensures that μ satisfies Condition *(i)*. Also note that in view of the simple consequence following the statement of Condition *(C)* we have for all $x \in V$ with $t(x) = \bullet$ that there are leaves $y', y'' \in L(x)$ with $\sigma(y') \neq \sigma(y'')$. Thus $\mathrm{lca}_S(\mu(L(x)) \in W \setminus B$, i.e. μ satisfies Condition *(ii)*. Also note that, by definition, alternative (M2) ensures that μ satisfies Condition *(v)*.

Claim: If $x, y \in G$ with $x \prec_T y$ then $\mu(x) \prec_S \mu(y)$.

Since y cannot be a leaf of T as $x \prec_T y$ we have $t(y) = \bullet$. There are two cases to consider, either $t(x) = \bullet$ or $t(x) = \odot$. In the latter case $\mu(x) = \sigma(x) \in B$ while $\mu(y) \in W \setminus B$ as argued above. Since $x \in L(y)$ we have $\mu(x) \prec_S \mu(y)$, as desired.

Now suppose $t(x) = \bullet$. Again by the simple consequence following Condition *(C)*, there are leaves $x', x'' \in L(x)$ with $a = \sigma(x') \neq \sigma(x'') = b$. Since $x \prec_T y$ and $t(y) = \bullet$, by Condition *(C)*, we conclude that $c = \sigma(y') \notin \sigma(L(x))$ holds for all $y' \in L(y) \setminus L(x)$. Thus, $((a,b),c) \in \mathfrak{S}$. But then $((a,b),c)$ is displayed by S and therefore $\mathrm{lca}_S(a,b) \prec_S \mathrm{lca}_S(a,b,c)$. Since this holds for all triples $((x',x''),y') \in \mathfrak{G}$ with $x', x'' \in L(x)$ and $y' \in L(y) \setminus L(x)$ we conclude $\mu(x) = \mathrm{lca}_S(\sigma(L(x))) \prec_S \mathrm{lca}_S(\sigma(L(x)) \cup \sigma(L(y) \setminus L(x))) = \mathrm{lca}_S(\sigma(L(y))) = \mu(y)$, establishing the claim.

It follows immediately that μ also satisfies Condition *(iv.2)* if x and y are contained in G.

Next, we extend the map μ to the entire vertex set V of T using the following observation. Let $x \in V$ with $t(x) = \blacksquare$. We know by Lemma 38 that $\mu(x)$ is an edge $[u,v] \in H$ so that $\mathrm{lca}_S(\sigma(L(x))) \preceq_S v$. Such an edge exists for $v = \mathrm{lca}_S(\sigma(L(x)))$ by construction. Every speciation vertex $y \in V$ with $x \prec_T y$

therefore necessarily maps above this edge, i.e., $u \preceq_S \mu(y)$ must hold. Thus we set

(M3) $\mu(x) = [u, \mathrm{lca}_S(\sigma(L(x)))]$ if $t(x) = \blacksquare$.

which now makes μ a map from V to $W \cup H$.

By construction, Conditions *(iii)*, *(iv.2)* and *(v)* are thus satisfied by μ. On the other hand, if there is a speciation vertex y between two duplication vertices x and x' of T, i.e., $x \prec_T y \prec_T x'$, then $\mu(x) \prec_S \mu(x')$. Thus μ also satisfies Condition *(iv.1)*.
It follows that μ is a reconciliation map from (T,t,σ) to S. □

Corollary 42. *Suppose that S is a species tree for (T,t,σ) and that L and B are the leaf sets of T and S, respectively. Then a reconciliation map μ from (T,t,σ) to S can be constructed in $O(|L||B|)$.*

Proof. In order to find the image of an interior vertex x of T under μ, it suffices to determine $\sigma(L(x))$ (which can be done for all x simultaneously, e.g. by bottom up transversal of T in $O(|L||B|)$ time) and $\mathrm{lca}_S(\sigma(L(x)))$. The latter task can be solved in linear time using the idea presented in [Zhang, 1997] to calculate the lowest common ancestor for a group of nodes in the species tree. □

We remark that given a species tree S on B that displays all triples in $\mathfrak{S}(T,t,\sigma)$, there is no freedom in the construction of a reconciliation map on the set $\{x \in V \mid t(x) \in \{\bullet, \odot\}\}$. The duplication vertices of T, however, can be placed differently, resulting in possibly exponentially many reconciliation maps from (T,t,σ) to S.

Lemma 39 implies that consistency of the triple set $\mathfrak{S}(T,t,\sigma)$ is necessary for the existence of a reconciliation map from (T,t,σ) to a species tree on B. Theorem 41, on the other hand, establishes that this is also sufficient. Thus, we have

Theorem 43. *There is a species tree on B for (T,t,σ) if and only if the triple set $\mathfrak{S}(T,t,\sigma)$ is consistent.*

We remark that a related result is proven in [Chauve and El-Mabrouk, 2009, Theorem.5] for the full tree reconciliation problem starting from a forest of gene trees.

It may be surprising that there are no strong restrictions on the set $\mathfrak{S}(T,t,\sigma)$ of triples that are implied by the fact that they are derived from a gene tree (T,t,σ).

Theorem 44. *For every set $\mathfrak{X}$ of triples on some finite set B of size at least one there is a gene tree $T = (V,E)$ with leaf set L together with an event map $t : V \to \{\bullet, \blacksquare, \odot\}$ and a map $\sigma : L \to B$ that assigns to every leaf of T the species in B it resides in, such that $\mathfrak{X} = \mathfrak{S}(T,t,\sigma)$.*

Proof. Irrespective of whether $\mathfrak{X}$ is consistent or not we construct the components of the required 3-tuple (T,t,σ) as follows: To each triple $r_k = ((x_{k1},x_{k2}),x_{k3}) \in \mathfrak{X}$ we associate a triple $T_k = ((a_{k1},a_{k2}),a_{k3})$ via a map $\sigma_k : L_k = \{a_{k1},a_{k2},a_{k3}\} \to \{x_{k1},x_{k2},x_{k3}\}$ with $\sigma(a_{ki}) = x_{ki}$ for $i = 1,2,3$ where we assume that for any two distinct triples $r_k, r_l \in \mathfrak{X}$ we have that $\sigma_k(L_k) \cap \sigma_l(L_l) = \emptyset$. Then we obtain $T = (V,E)$

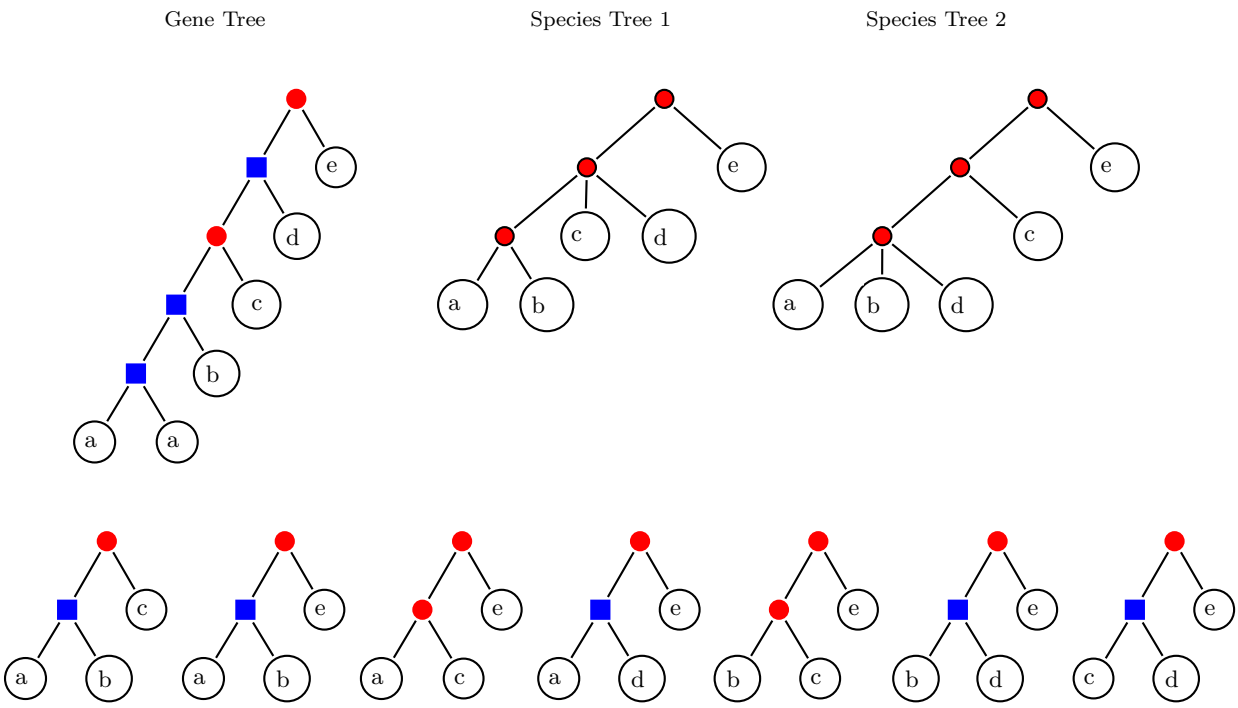


Figure 7.4: The set $\mathfrak{S}(T,t,\sigma)$ inferred from the event labeled gene tree (T,t,σ) does not necessarily define a unique species tree. For clarity of exposition, we have identified, via the map σ, the leaves of the gene tree and of the set of triples $\mathfrak{S}(T,t,\sigma)$ with the species they reside in .

by first adding a single new vertex ρ_T to the union of the vertex sets of the triples T_k and then connecting ρ_T to the root ρ_k of each of the triples T_k. Clearly, T is a phylogenetic tree on $L = \dot{\bigcup}_{r_k \in \mathfrak{X}} L(\rho_k)$. Next, we define the map $t : V \to \{\bullet, \blacksquare, \odot\}$ by putting $t(\rho_T) = \blacksquare$, $t(a) = \odot$ for all $a \in L$ and $t(a) = \bullet$ for all $a \in V - (L \cup \{\rho_T\})$. Finally, we define the map $\sigma : L \to B$ by putting, for all $a \in L$, $\sigma(a) = \sigma_k(a)$ where $a \in L_k$. Clearly $\mathfrak{S}(T,t,\sigma) = \mathfrak{X}$. □

We remark that the gene tree constructed in the proof of Theorem 44 can be made into a binary tree by splitting the root ρ_T into a series of duplication and loss events so that each subtree is the descendant of a different paralog.

Since by Theorem. 44 there are no restrictions on the possible triple sets $\mathfrak{S}(T,t,\sigma)$, it is clear that S will in general not be unique. An example is shown in Fig.7.4.

7.3 Results for simulated species and event-labeled gene trees

In order to determine empirically how much information on the species tree we can hope to find in event labeled gene trees, we simulated species trees together with corresponding event-labeled gene trees with different duplication and loss rates. The process to generate these trees will be described in the following chapter.

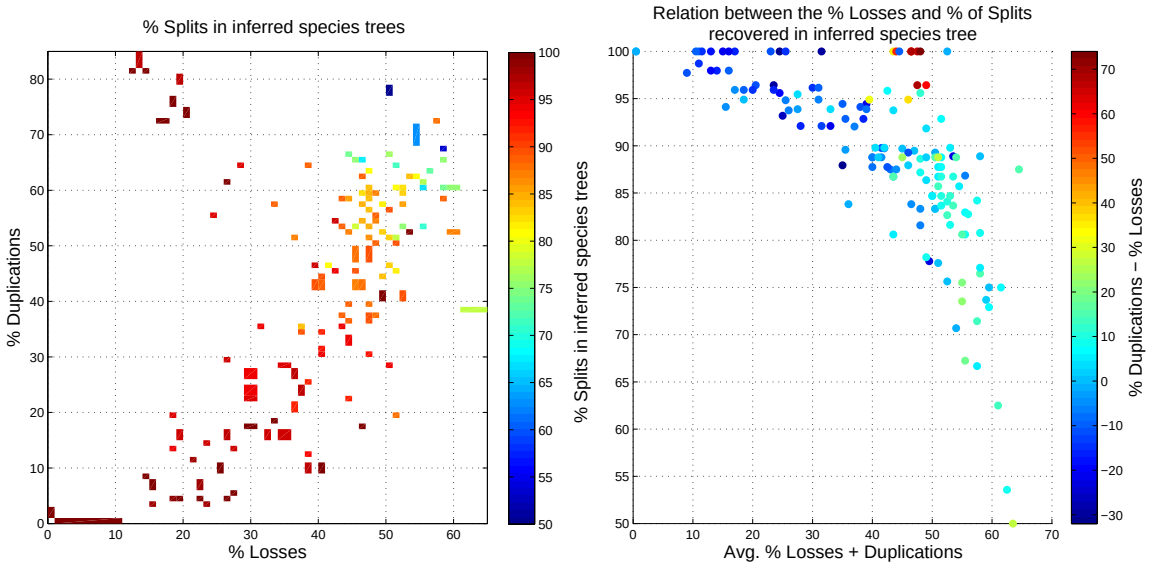


Figure 7.5: **Left:** Heat map that represents the percentage of recovered splits in the inferred species tree from triples obtained from simulated event-labeled gene trees with different loss and duplication rates.
Right: Scattergram that shows the average of losses and duplications in the generated data and the accuracy of the inferred species tree.

Approximately 150 species trees with 10 to 100 species were generated. For each species tree, we then simulated a gene tree. After determining the triple set $\mathfrak{S}(T,t,\sigma)$ according to Theorem 41, we used `BUILD` to compute the species tree. In all cases `BUILD` returns a tree that is a homomorphic contraction of the simulated species tree. The difference between the original and the reconstructed species tree is thus conveniently quantified as the difference in the number of interior vertices. Note that in our situation this is the same as the split metric [Semple and Steel, 2003].

The results are summarized in Fig. 7.5. Not surprisingly, the recoverable information decreases in particular with the rate of gene loss. Nevertheless, at least 50% of the splits in the species tree are recoverable even at very high loss rates. For moderate loss rates, in particular when gene losses are less frequent than gene duplications, nearly the complete information on the species tree is preserved. It is interesting to note that `BUILD` does not incorporate splits that are not present in the input tree, although this is not mathematically guaranteed.

7.4 Concluding Remarks

Event-labeled gene trees can be obtained by combining the reconstruction of gene phylogenies with methods for orthology detection. Orthology alone already encapsulates partial information on the gene tree. More precisely, the orthology relation is equivalent to a homomorphic image of the gene tree in which adjacent vertices denote different types of events. We discussed here the properties of

reconciliation maps μ from a gene tree T along with an event labelling map t and a gene to species assignment map σ to a species tree S. We show that (T,t) event labeled gene trees for which a species tree exists can be characterized in terms of the set $\mathfrak{S}(T,t,\sigma)$ of triples that is easily constructed from a subset of triples of T. Simulated data shows, furthermore, that such trees convey a large amount of information on the underlying species tree, even if the gene loss rate is high.

It can be expected that for real-life data the tree T contains errors so that $\mathfrak{S} := \mathfrak{S}(T,t,\sigma)$ may not be consistent. In this case, an approximation to the species tree could be obtained e.g. from a maximum consistent subset of $\mathfrak{S}$. Although (the decision version of) this problem is NP-complete [Jansson, 2001, Wu, 2004], there is a wide variety of practically applicable algorithms for this task, see [He et al., 2006, Byrka et al., 2010a]. Even if $\mathfrak{S}$ is consistent, the species tree is usually not uniquely determined. Algorithms to list all trees consistent with $\mathfrak{S}$ can be found e.g. in [Ng and Wormald, 1996, Constantinescu and Sankoff, 1995]. A characterization of triple sets that determine a unique tree can be found in [Bryant and Steel, 1995]. Since our main interest is to determine the constraints imposed by (T,t,σ) on the species tree S, we are interested in a least resolved tree S that displays all triples in $\mathfrak{S}$. The `BUILD` algorithm and its relatives in general produce minor-minimal trees, but these are not guaranteed to have the minimal number of interior nodes. Finding a species tree with a minimal number of interior nodes is again a hard problem [Jansson et al., 2012]. At least, the vertex minimal trees are among the possibly exponentially many minor minimal trees enumerated by Semple's algorithms [Semple, 2003]. For more details, refer to Chapter 4 where we have discussed more widely about this problems.

For a given species tree S, it is rather easy to find a reconciliation map μ from (T,t,σ) to S. A simple solution μ is closely related to the so-called LCA reconcilation: every node x of T is mapped to the last common ancestor of the species below it, $\mathrm{lca}_S(\sigma(L(x)))$ or to the edge immediately above it, depending on whether x is speciation or a duplication node. While this solution is unique for the speciation nodes, alternative mappings are possible for the duplication nodes. The set of possible reconciliation maps can still be very large despite the specified event labels.

If the event labeling t is unknown, there is a reconciliation from any gene tree T to any species tree S, realized in particular by the LCA reconciliation, see e.g. [Chauve and El-Mabrouk, 2009, Doyon et al., 2009]. The reconciliation then defines the event types. Typically, a parsimony rule is then employed to choose a reconciliation map in which the number of duplications and losses is minimized, see e.g. [Guigó et al., 1996, Bonizzoni et al., 2005, Burleigh et al., 2009, Górecki and J., 2006]. In our setting, on the other hand, the event types are prescribed. This restricts the possible reconciliation maps so that the gene tree cannot be reconciled with an arbitrary species tree any more.

Since the observable events on the gene tree are fixed, the possible reconciliations cannot differ in the number of duplications. Still, one may be interested in reconciliation maps that minimize the number of loss events. An alternative is to maximize the number of duplication events that map to the same edge in S to account for whole genome and chromosomal duplication events [Burleigh et al., 2009].

Our approach to the reconciliation problem via event-labeled gene trees opens up some interesting new avenues to understanding orthology. In particular, the results in this contribution combined with those in [Hellmuth et al., 2013] concerning cographs should ultimately lead to a method for automatically generating orthology relations that takes into account species relationships without having to explicitly compute gene trees. This is potentially very useful since gene tree estimation is one of the weak points of most current approaches to orthology analysis.

CHAPTER 8

Simulation of gene family histories and its applications

The reconstruction of the evolutionary history of large gene families has remained a hard and complex problem, which amounts to disentangling speciation events from gene duplication events. The evaluation of reconstruction algorithms is hampered, by the lack of well-studied cases that could serve as a gold standard. We present here a simulation environment designed to generate large gene families with complex duplication histories on which reconstruction algorithms can be tested and software tools can be benchmarked.

We use these simulations to test the accuracy of orthology predictions made by `OMA`, `OrthoMCL`, `Proteinortho` and the extended version of it: `PoFF`. We also present a method based on *forbidden subgraphs*, induced subgraphs on five vertices which contain more than one P_4, to mesure how good the prediction of orthology relations among sets of genes is.

8.1 Simulation of gene family histories

The way gene families and genomes evolve can be understood in detail only when the location of gene duplication episodes in the tree of life can be deciphered. Since most genes belong to larger gene families, the analysis of the gene family histories thus plays an important role in the study of genome evolution. Empirically, one frequently observes that the tree that describes the evolution of species, the species tree, is inconsistent with the tree that is obtained from a group of genes of a gene family (the gene tree). Goodman et al. [1979] deduced that this inconsistency might be the result of mistaking paralogs for orthologs. Orthologous genes refer to copies of genes that reveal the phylogeny of species, while paralogous genes have been created by duplication events. Phylogeny reconstruction can help to understand how gene families evolved and to identify the chronology of

duplications within a gene family of a single species. There is, however, lack of both test data and evaluation procedures to test, compare, and benchmark the performance and results of prediction tools and methods. Here we present an efficient method that simulates phylogenetic processes that fulfills those needs.

The simulation of gene family histories starts with the generation of a *species tree*. Within this rooted bifurcating tree the nodes represent species and edges their relation. Specifically, internal nodes represent ancient species whereas leaf nodes represent extant species. Given a number N of species, we generate a random species tree S under the "Age Model" described in [Keller-Schmidt et al., 2010]. This model starts with a rooted tree with two leaves. In an iterative process one of the leaves is selected and two new leaves are attached to it until the tree has N leaves. This model makes use of the idea that the longer a node has not been involved in a speciation, the less likely it will be in the future. These trees are balanced and edge lengths (which represent time) are normalized so that the total length of the path from the root to each leaf is 1.

For each species tree S, we then simulated gene trees using the following rules:

1. The root of S contains an ordered list of ancestral genes, one for each gene family. The number of families is a user-defined parameter.

2. S is traversed in a depth first order. All changes to the genome are simulated independently for each edge of S with constant rates.

3. At each internal node of S, the ordered gene list received from its parental edge is copied without change to both offspring edges.

4. Along each edge of S a number of events is sampled from a stochastic Poisson Process $P_{\lambda,l}$, where the parameter $\lambda \in [0,1]$ is the probability of the event to happen and l is the branch length. The process may generate none, one, or several events of the following types:

   - Gene duplication: one gene gets duplicated.
   - Cluster duplication: a group of consecutive genes gets duplicated.
   - Genome duplication: the whole group of genes gets duplicated.
   - Gene loss: A gene gets lost and therefore removed from the genome.

   In the case of duplications the probability of the events to happen is inversely proportional to the number of genes in the organism, so that, the larger the familiy is the smaller the probability is to generate new gene copies.

5. A special rule applies to recently duplicated genes to account for the deletion of redundant gene copies before they can be stabilized by sufficient functional divergence or subfunctionalization [Ohno, 1999, Lynch and Conery, 2000]. We model this by a probability $\theta = l_1 + P \times l_2$ of immediate loss where P is the size of the gene family immediately after the duplication event, this will allow the user to enlarge or contract families by adjusting l_1 and l_2, while taking into

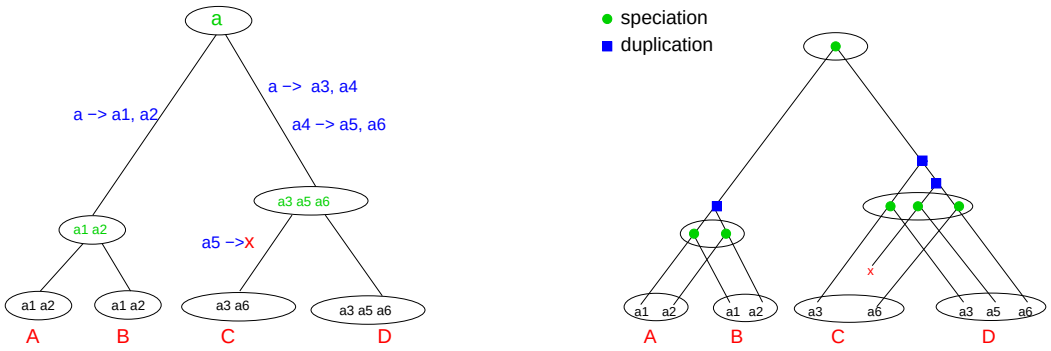


Figure 8.1: **(a)** A one-gene family history: from a node parent to a node child, there could be duplications and losses of genes. **(b)** The reconciled tree: the gene tree embedded in the species tree. Each internal node represents an event, either an speciation or a gene duplication.

account the gene family size. This model in particular accounts for the increased loss rates in the wake of multiple gene duplications and in particular for genome duplications [Prohaska et al., 2004]. These losses are constrained so that at least one copy of the gene family is retained.

6. To obtain an order of the generated genes, rearrangements are carried out for each edge of S using translocation and inversion operations on the ordered list of genes that "survived" until the next speciation. Breakpoints are picked randomly and the number of inversion operations is chosen uniformly proportional to the branch length [Xu et al., 2007].

The result of this simulation is a gene tree T_i for each family i together with a true reconciliation map to the species tree S. All gene lineages terminating in a deletion event are pruned from the gene tree so that we retain a gene tree T_i in which only extant genes appear as its leaves. The known reconciliation furthermore provides us with a labeling of the internal nodes of T_i with *duplication* or *speciation* events, see Figure 8.1. This in turn determines the true orthology relation for all genes received in the leaves of S. In addition to that, the gene orders within their respective genomes is obtained.

Additionally, the algorithm can generate one gene tree for each species, i.e. the pruned reconciled tree containing only genes of a certain species. Furthermore, for each gene family the orthology and homology matrices are computed. To generate the orthology matrix, we say that two genes are orthologous if their lowest common ancestor (LCA) in the reconciled tree represents a speciation event. To generate the homology matrix, a gene a from species i is homologous to gene b from species j if for every gene c from species i and every gene d from species j the $LCA(a,b) \leq LCA(c,b)$ and $LCA(a,b) \leq LCA(a,d)$.

8.2 How close is the induced graph by a given orthology relation to a cograph?

A graph G is P_4-*sparse* if every subset of $V(G)$ with five vertices induces at most one P_4.

An interesting feature of P_4-sparse graphs is that they generalize the cographs and therefore, they admit a tree representation unique up to isomorphism [Jamison and Olariu, 1992].

Forbidden subgraphs we call those induced subgraphs on five vertices which contain more than one P_4. Given that a valid orthology relation can be characterized as a cograph, it has been shown that P_4-sparse graphs can be edited by adding and removing edges to obtain a cograph in polynomial time [Liu et al., 2011]. For a graph that is not P_4-sparse, a solution for the cograph editing problem is *NP*-complete [Liu et al., 2011].

In this section we describe a method to measure how close the induced graph by orthology relation is to a cograph, in terms of its P_4-sparseness and induced forbidden subgraphs. We use simulations like the ones described in the previous section to quantify the "noise" in a graph in comparison with random graphs.

8.2.1 P_4 Sparse Graphs

Cographs are in the class of P_4-sparse graphs, for the set of graphs which are P_4-sparse but not cographs, there is a polynomial time solution for the so called the *cograph editing problem*.

Definition 45. *(P4-sparse graphs, [Hoang, 1985]). A graph G is P4-sparse if every induced subgraph* $H \subseteq G$ *with* $|V(H)| = 5$ *contains at most one induced P4.*

Liu et al. [2011] presented the `EDP4` algorithm which takes a P4-sparse graph G as input and outputs a minimal edge edition set to convert G into a cograph. The algorithm decomposes G into connected components, single vertices and spider graphs, and it is based on the following lemma:

Lemma 46. *[Liu et al., 2011]. For a graph G the following conditions are equivalent:*

1. *G is a* P_4*-sparse graph.*
2. *For every induced subgraph H of G with at least two vertices, exactly one of the following statements is satisfied:*
   (a) *H is disconnected*
   (b) $\overline{H}$ *is disconnected*
   (c) *H is a spider*

where *spider graphs*, illustrated in Fig. 8.2, are a basic component of P_4-sparse graphs. These graphs have a special structure which contain only one induced P_4 on 5 vertices, and which is easy to identify (for more details on this type of graph refer to [Jamison and Olariu, 1992]).

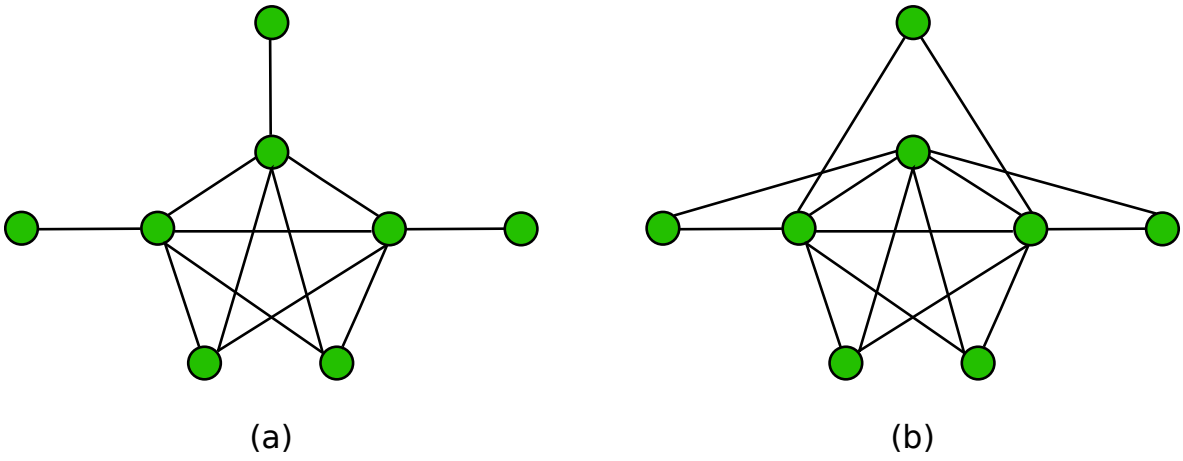


Figure 8.2: Spider graphs.**(a)** Thin spider. **(b)** Thick spider.

8.2.2 Forbidden Subgraphs

The definition of P_4-sparse graphs leads us to the observation that for such graphs, it holds, that they do not contain any of the forbidden subgraphs shown in Fig. 8.3. Each of these subgraphs consists of five vertices and includes more than one induced P_4, implying that these subgraphs cannot be contained in P_4-sparse graphs. Let H be one of the forbidden subgraphs, it follows that both H and its complement are forbidden subgraphs. This can be observed in Fig. 8.3, where the prefix of some subgraphs' name denotes that this is the complement of the subgraph with the corresponding name. For the graphs P_5, *kite*, *fork*, their complement are *co-*P_5, *co-kite*, *co-fork*, respectively. For the C_5 it can easily be checked that its complement is as well a C_5.

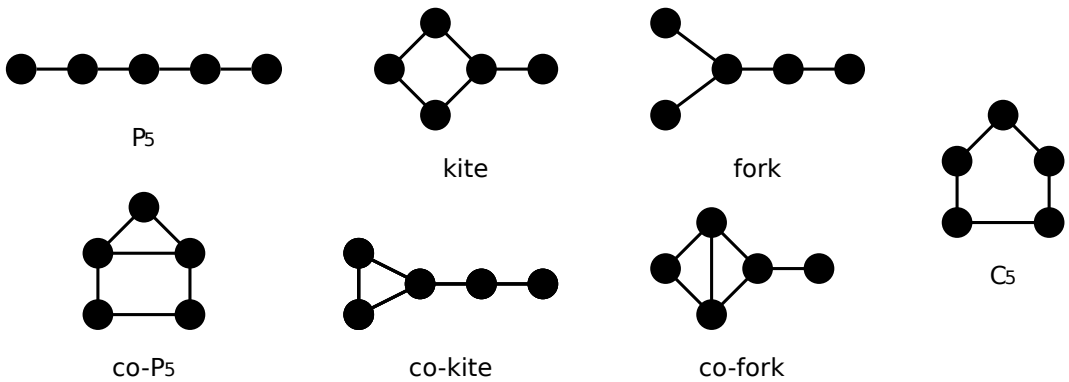


Figure 8.3: The seven forbidden subgraphs that are not contained in a P_4-sparse graph.

Liu et al. [2011] proved that for a graph which is neither P_4-sparse nor a cograph the problem is *NP*-complete, but fixed- parameter tractable when delimiting the number of edges to edit.

Berkemer [2012] presented a pipeline for estimated orthology relations to be converted into cograpghs as well as improvements in the running time for the cograph editing problem when fixed-parameter

tractable (where the parameter is the number of edges to be edited) by using modular decomposition described by Habib et al. [2004].

In this chapter we are interested in quantifying forbidden subgraphs, P_4-sparse subgraphs and cographs in valid orthology relations and in random graphs and in how good an estimated orthology relation is so as to be able to assess. In other words, we would like to be able to differentiate between graphs that can be edited and converted to cographs (which in turn would give us a valid orthology relation) and graphs that look more like random graphs and therefore would not be worth trying to convert to cographs.

8.2.3 Induced subgraphs in simulated data and in random graphs

We simulated gene trees as described in Section 8.1. A series of species trees with N leaves, with $N \in [5, 100]$ are generated. For each generated species tree S, one gene tree T is generated as well. We set the following parameteres for the simulation of these trees:

1. Probability of gene duplication = 0.9
2. Probability of cluster duplication = 0.3
3. Probabiliy of genome duplication = .001
4. Probability of gene loss = 0.5
5. Parameters l_1 and l_2 for the probability $\theta = l_1 + P \times l_2$ to account for gene losses after genome duplications are both set to 0.1.

The internal nodes of each gene tree are labeled with "duplication" or "speciation" events and therefore we are able to obtain the corresponding orthology matrix as M. In particular, this matrix represents a graph $G = (V, E)$, where V is the set of leaves in the gene tree and an edge $\{a, b\} \in E$ if $M(a, b) = 1$. By definition, G will be a cograph.

We perturbed the cographs by adding a percentage of "noise". The idea here is to find out the point where the noisy cograph becomes indistinguishable from a random graph. The introduction of "noise" in a cograph is carried out by removing edges and adding new ones while preserving vertex degrees. For a pair of edges $\{a, b\}$ and $\{c, d\}$, these are removed from the graph while introducing the pair of edges $\{a, c\}$ and $\{b, d\}$ or the pair $\{a, d\}$ and $\{b, c\}$, if none of these already exist in the original graph. Then here "the percentage of noise" is defined as the percentage of edges in the original graph to be edited. It is crucial to note that there is a very important restriction: new edges can be introduced only if they connect vertices that represent genes from different species.

In our simulations, we introduce from 1% to 95% of noise to each set of cographs obtained from the gene trees with different number of species. Therefore, there are 96 "noisy" graphs for each percentage of noise. Noise from 1% to 20% was incremented in steps of 1. Noise from 25% to 95% was incremented in steps of 5.

For each noisy graph, we then obtained the number of vertices and generated random graphs with the same number of vertices and same number of edges. Random graphs are generated in such a way that vertex degrees have the same distribution that the corresponding noisy graph. To ensure this we implemented the following procedure:

1. A graph with the same number of vertices as the noisy graph without edges is created.
2. For each noisy graph, the degree sequence is obtained and ordered in non-increasing order.
3. The first vertex a with degree n is connected to the next n vertices in the degree sequence.
4. Vertex a is removed from the degree sequence and this is reordered.
5. This process is repeated until all vertex degrees are fulfilled.

It is important to mention that in random graphs, we have no species assignment to each vertex of the graph, therefore, there is no restriction when connecting two vertices as the one with noisy graphs.

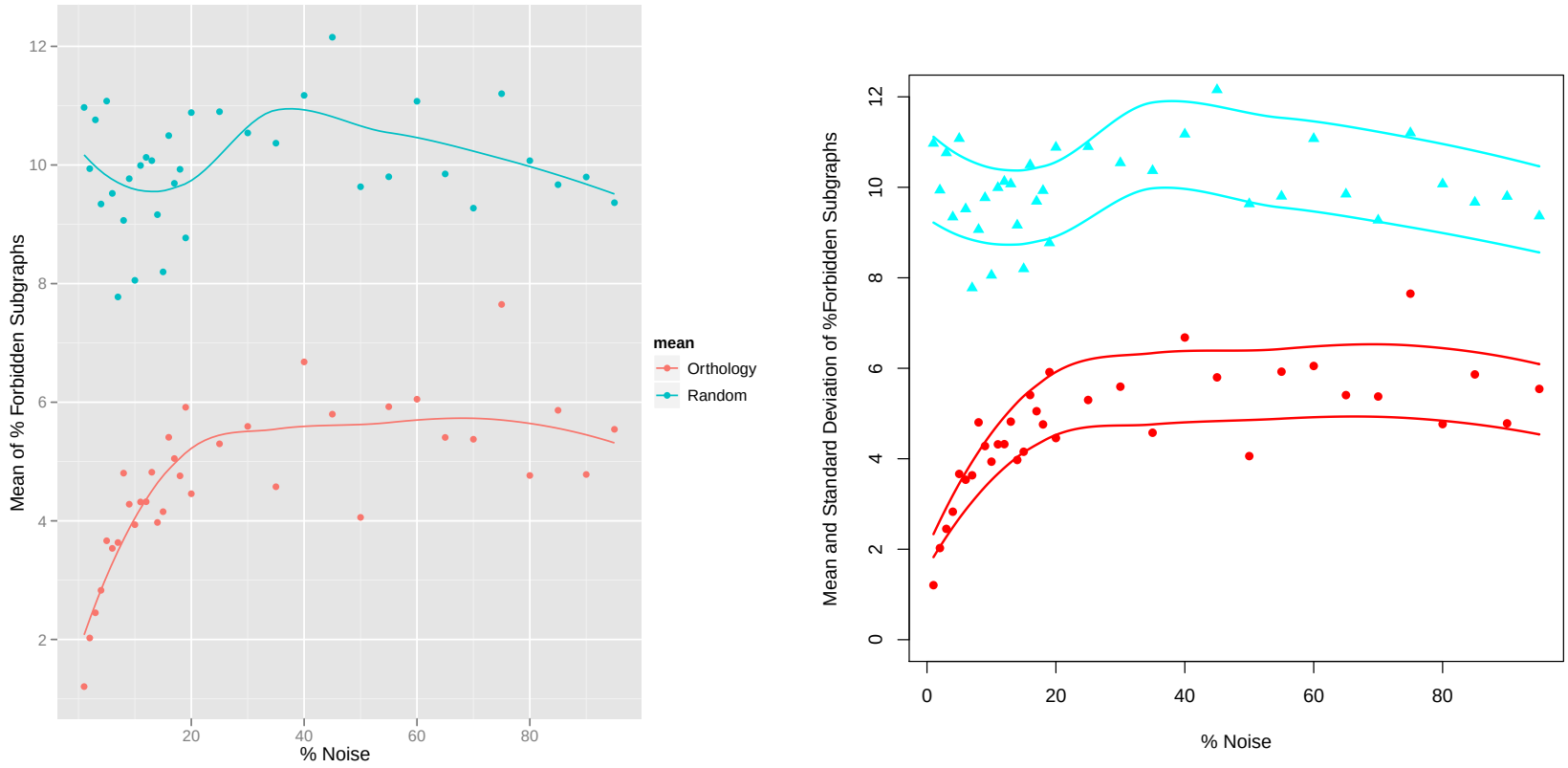
For each simulated noisy graph and the corresponding random graph, we took a sample of 1000 induced subgraphs on five vertices and each was classified in one of the four categories:

- Forbidden
- P_4-sparse
- Cograph
- Small Connected Component

The algorithm that samples induced subgraphs on 5 vertices, takes 5 random vertices and obtains the induced subgraph. As each graph can contain several connected components, the algorithm makes sure that once it has picked a vertex from a connected component in the graph, it will pick the other 4 vertices in the same connected component. If the connected component contains less than 5 vertices, the induced subgraph will be reported as small connected component. For each set of 1000 samples, the percentages of forbidden, P_4-sparse, cographs and small connected components are calculated.

Finally, for each set of noisy graphs with a specific percentage of noise and the corresponding random graph, we obtained the mean and the standard deviation for each category. Figs. 8.4-8.6 show the result of these simulations. We have omitted the results for small connected components since there were hardly any of them in our simulated data.

As can be observed, the mean of the number of induced forbidden subgraphs, P_4-sparse graphs and cographs in random graphs is independent of the numbers of vertices and edges. We can also observe that cographs with 20% or more of noise converge to a certain range and therefore become very similar to those of random graphs. Here, the statistics on the means of random graphs and noisy graphs for each category do not really converge, this is due to the restriction imposed to noisy graphs.



(a) (b)

Figure 8.4: For each percent of noise, 96 sets of orthology relations were simulated with corresponding random ones. Random graphs with same number of vertices and same vertex degree. **(a)** Means of forbidden subgraphs for noisy graphs and for random graphs are represented with red dots and blue dots, respectively. **(b)** Means as dots and standard deviations as error rate as curves show the behaviour of noisy cographs in comparison with random graphs. Graphs with more than 20% of noise converge to a certain range which makes them indistinguishable from random graphs. However, graphs with less than 20% of noise can be edited and converted to cographs.

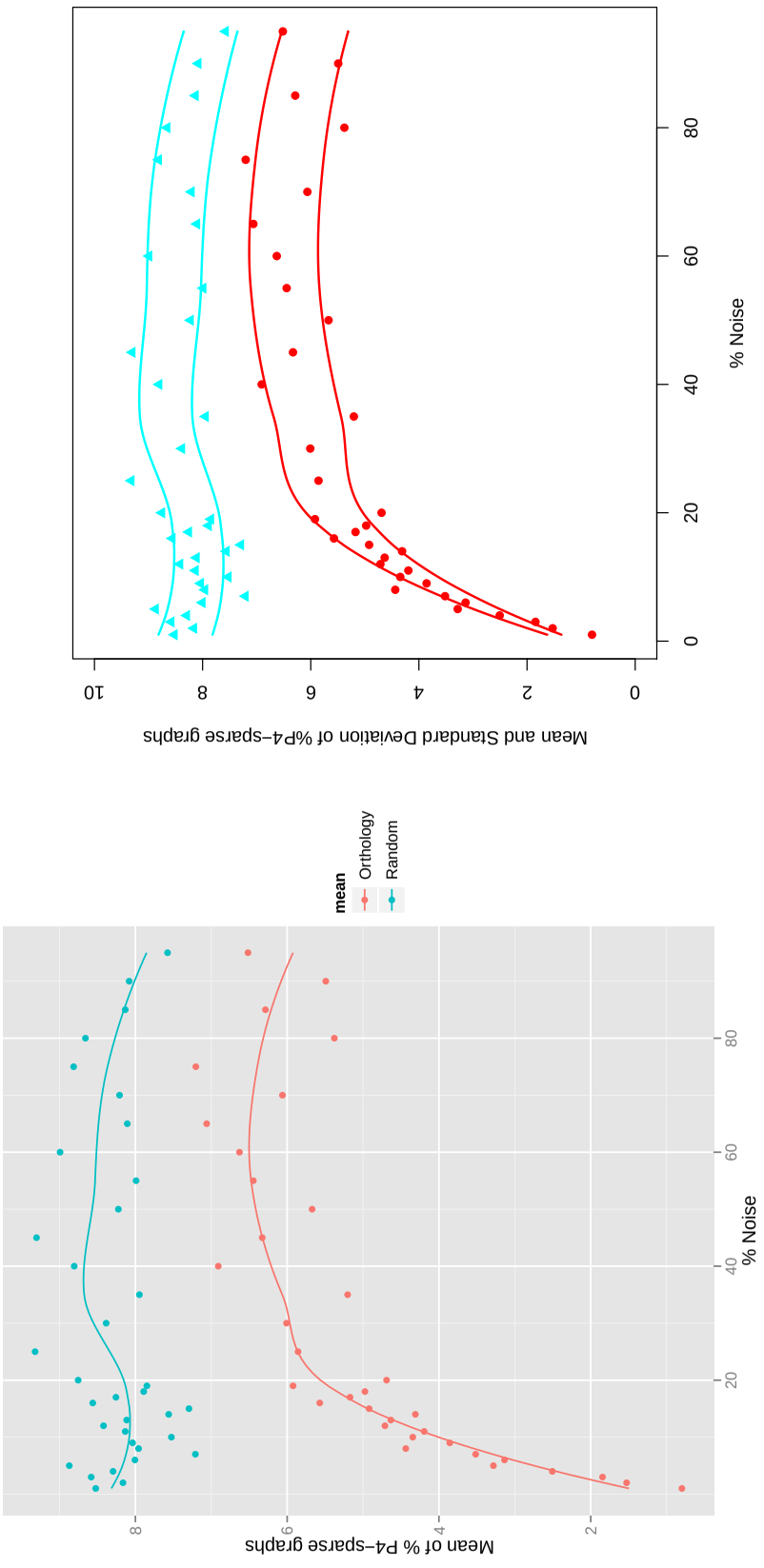
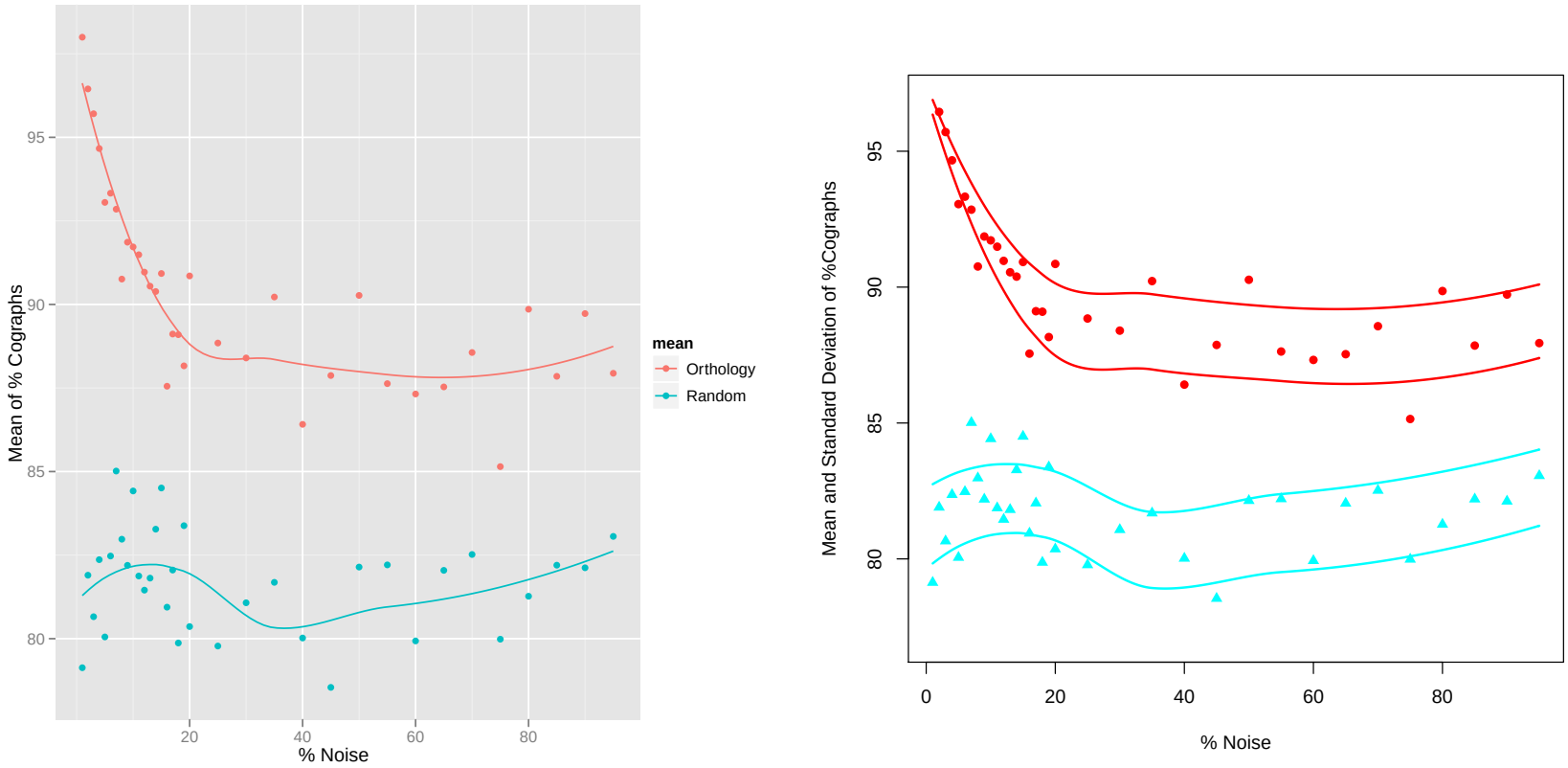


Figure 8.5: For each percent of noise, 96 sets of orthology relations were simulated with corresponding random ones. Random graphs with same number of vertices and same vertex degree. **(a)** Means of induced P_4-sparse graphs in five vertices for noisy graphs and for random graphs are represented with red dots and blue dots, respectively. **(b)** Means as dots and standard deviations as error rate as curves show the behaviour of noisy cographs in comparison with random graphs. Graphs with more than 20% of noise converge to a certain range which makes them indistinguishable from random graphs. However, graphs with less than 20% of noise can be edited and converted to cographs.



(a) (b)

Figure 8.6: For each percent of noise, 96 sets of orthology relations were simulated with corresponding random ones. Random graphs with same number of vertices and same vertex degree. **(a)** Means of induced Cographs in five vertices for noisy graphs and for random graphs are represented with red dots and blue dots, respectively. **(b)** Means as dots and standard deviations as error rate as curves show the behaviour of noisy cographs in comparison with random graphs. Graphs with more than 20% of noise converge to a certain range which makes them indistinguishable from random graphs. However, graphs with less than 20% of noise can be edited and converted to cographs.

In Figure 8.7 we can observe the relation between forbidden subgraphs and induced P_4-sparse graphs in noisy graphs. The first observation is that independently of the percentage of noise introduced to the cographs, the number of these two types of subgraphs increases almost directly proportionally. However, when looking at Figure 8.7(a), we can note that with a low level of noise, the high rate (close to 100%) of induced cographs is conserved, sometimes even with a percentage of noise close to 20. Of course, after the introduction of more than 25% of noise, many more P_4-sparse subgraphs are created.

However, when observing Figure 8.8, it all becomes more interesting. When comparing Figure 8.8(a) and Figure 8.8(b), we can see that with 1% to 20% of noise, there is a bias towards more forbidden subgraphs than with more than 20% of noise, this can be explained by the observation that when adding or removing one edge in a cograph, it is possible to create one or more P_4's, therefore when we have a predefined graph structure such as a cograph, this structure gets destroyed as more and more edges are edited. However, a graph that has more than 20% of noise, already resembles a random graph and thus, the editing of edges does not much affect its structure.

With these results we are able to quantify the "level" of noise that a given orthology relation has, and therefore find out whether it makes sense to try to edit the graph to get close to a cograph or whether when quantifying the number of subgraphs on 5 vertices, it shows that the graph looks like a random graph and therefore the orthology relation is likely to be incorrect.

8.2.4 Real Data: measuring noise in OMA

In the OMA database [Altenhoff et al., 2011], the identification of orthologs among available complete genomes has been carried out. This database includes more than 1000 genomes so far, and one is able to find pairs of orthologs for any two genomes. We have downloaded their predictions and obtained the corresponding orthology relation graph. We have filtered the data to obtain only orthology relations of the type: many-to-many, one-to-many or many-to-one. Since the type 1-to-1 will be seen as a connected component of size two and therefore unable to create P_4's we decided to exclude them to speed the calculation of induced subgraphs on five vertices.

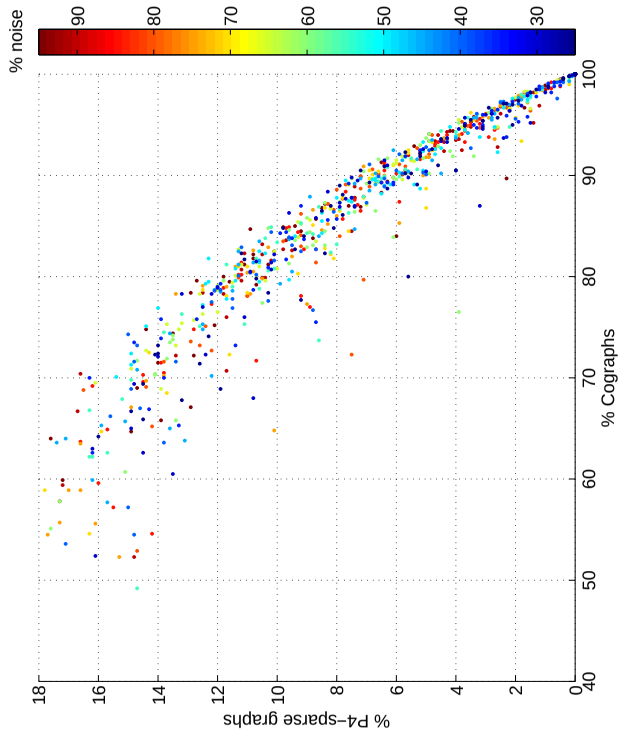
We test two datasets, prokaryotes and eukaryotes, to measure the accuracy of the orthology relations based on the forbidden subgraphs, induced P_4-sparse graphs, cographs and small connected components. As these data sets are large, we also tested a smaller dataset of some eukaryotes that are more closely related, the eight flowering plants: rice, sorghum, maize, cassava, poplar, wheat, arabidopsis and grape, plus an alga.

The results are shown in Table 8.1. Here we can observe that the graph corresponding to the orthology relation contains a lower percentage of forbidden subgraphs and induced P_4-sparse graphs and on the other hand, of about 90% of induced cographs, therefore it must be worth trying to apply a cograph editing method to get rid of the false orthology edges in the graphs. It is worth mentioning that the total number of small connected components are calculated for each dataset, and consists of the 30% to 40% of the total number of connected components. However, none of these small

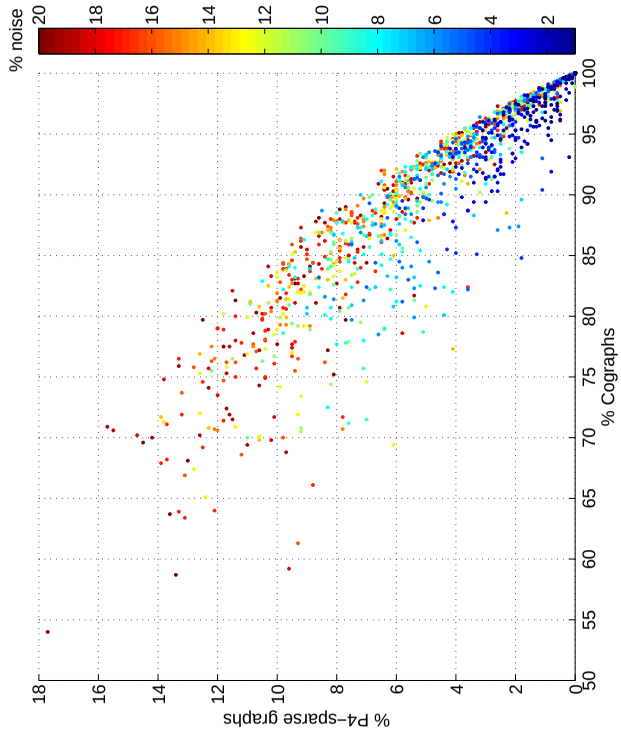
connected components forms a clique. Taking into account that we removed all isolated edges, we are left with connected components of three and four genes, the latter type might then be a P_4 and therefore a P_4-sparse subgraph that can be edited to a cograph.

In Table 8.1 and in all the tables presented in the reminder of this chapter, we use the following notation:

- Let $G(V,E)$ be the induced graph obtained from the orthology relation predicted by a certain method.
- $|V|$ and $|E|$ are the sizes of the sets of vertices and edges in the graph, respectively.
- $|S|$ is the number of species involved.
- $|CC|$ is the number of connected components found in G.
- $|smallCC|$ is the number of connected components that contain less than five vertices.
- $|K_s|$ is the number of cliques of size s.
- %Forbs is the abbreviation for the percentage of induced forbidden subgraphs on five vertices.
- $\%P_4$-s is the abbreviation for the percentage of induced P_4-sparse graphs on five vertices.
- %Cographs is the percentage of induced cographs on five vertices.



(b)



(a)

Figure 8.7: Relation of cographs and P_4-sparse graphs with 5 to 100 species. The percentage of cographs grows more less directly proportional to the percentage of P_4-sparse graphs (**a**) Graphs with 1% to 20% of noise conserve most of the cograph structure. (**b**) Graphs with 25% to 95% of noise tend to contain more P_4-sparse graphs.

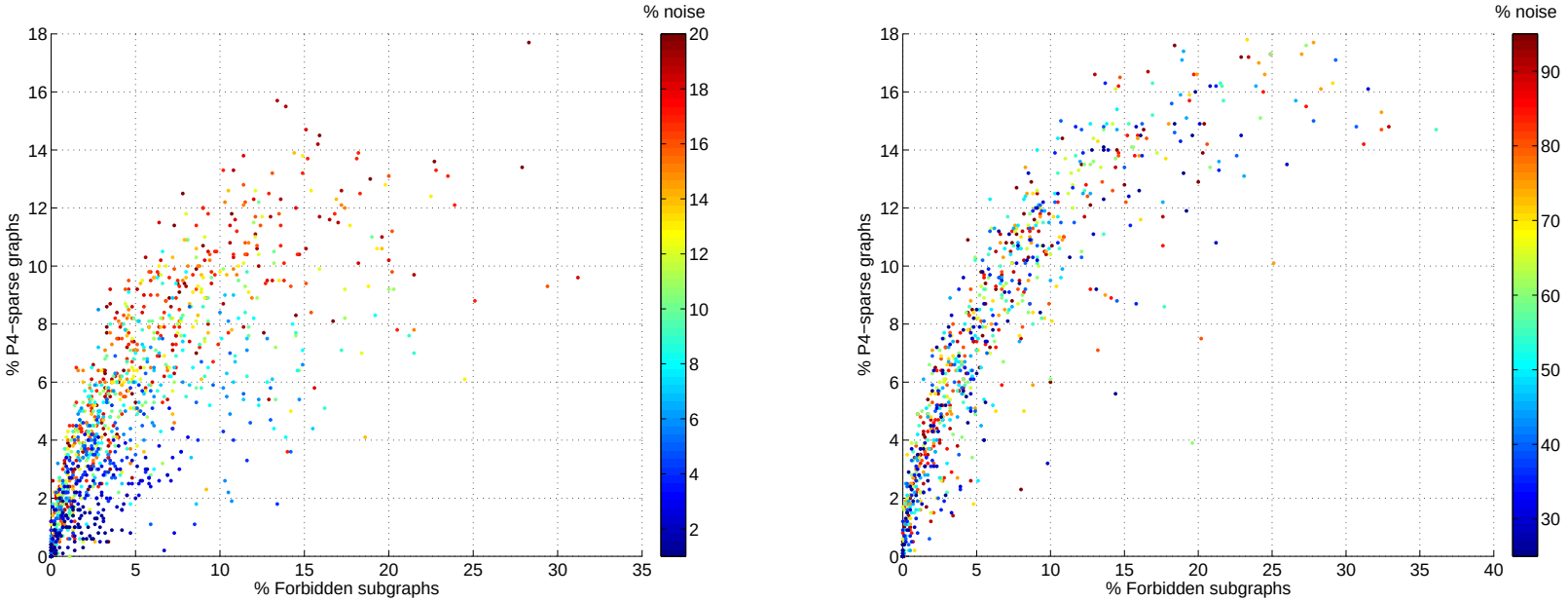


Figure 8.8: Relation of forbidden subgraphs and P_4-sparse graphs with 5 to 100 species. **(a)** Graphs with 1% to 20% are biased towards more forbidden subgraphs, since the editing of an edge can create none, one or more P_4's the cograph structure is destroyed as more edges are edited. **(b)** Graphs with 25% to 95% of noise are more similar to random graphs, therefore the editing of edges does not really affect their structure.

Dataset	$\|S\|$	$\|V\|$	$\|E\|$	$\|CC\|$	$\|smallCC\|$	%Forbs	$\%P_4$-s	%Cographs
Prokaryotes	1076	2268188	72262333	19077	5273	5.74	1.52	92.74
Eukaryotes	135	1173438	13482684	23589	7913	6.46	1.82	91.72
Flowering Plants	9	152140	583919	12480	3368	8.65	3.07	88.28

Table 8.1: Measuring noise in the OMA database: 1 million samples of induced subgraphs were taken for each dataset. Flowering plants are: rice, sorgan, maiz, yuca, puttler tree, weath, milk weather, grape and a primitive plant which is an algi. Small connected components do not form cliques and therefore some of those with four vertices might form a P_4. However, the low percentage of forbidden subgraphs found shows that this database contains realiable ortholoy predictions and therefore, it may be easy to find the wrong predictions by using a cograph editing method.

8.3 Application: Testing BBH, Proteinortho, PoFF and OrthoMCL

The bidirectional best hit (BBH) criterion is often used to identify orthologs for pairs of genomes. This method requires that for a candidate pair of orthologs a, b, that a is the best hit for b and vice versa, that b is the best hit for a. Proteinortho and OrthoMCL are commonly used tools when aiming to predict orthology in a group of genes. They both use a similar approach to BBH but have an additional clustering step as described in Table 3.2. PoFF [Lechner et al., 2013] is the extended version of Proteinortho which takes into account synteny information. It incorporates the heuristic method FFAdj-MCS described in [Doerr et al., 2012], which assesses pairwise gene order using conserved adjacencies and calculating a matching whose objective function maximizes for a trade-off between adjacencies and similarity scores of genes.

We have simulated data as described in section Section 8.1 for which the entire history and hence the orthology relation is known, to estimate how PoFF performs compared to the original Proteinortho implementation and to compare both with BBH and OrthoMCL. We also simulated sequence evolution and genomic rearrangements for four example data sets comprising 20, 50, 80 and 100 gene families in 20 species. All test sets feature duplications of both individual genes and gene clusters. The set with 80 gene families in addition includes whole genome duplications.

The parameters used for the simulation of these dataset are set as following:

1. Probability of gene duplication = 0.9
2. Probability of cluster duplication = 0.5
3. Probabiliy of genome duplication = 0
4. Probability of gene loss = 0.5
5. Parameters $l_1 = 0.3$ and $l_2 = 0$ for the probability $\theta = l_1 + P \times l_2$

The probability of genome duplication is set to 0.03 to generate genome duplications in the set with 80 gene families. We used indel-Seq-Gen [Strope et al., 2009] to generate simulated amino acid sequences for the simulated gene trees. After applying BBH, Proteinortho, PoFF and OrthoMCL to the datasets, we obtain the corresponging orthology graphs and sampled subgraphs as we did for the OMA database in Section 8.2.4.

When analysing the results, we realized that the orthology graph obtained by the output from OrthoMCL contains edges between pairs of genes that belong to the same species, thus violating the defintion of orthology. For example, from the total number of edges in the dataset with 80 gene families, more than 80% of them connect genes from the same species, and this might lead to dense graphs that could be close to cliques and therefore might look like graphs that are close to cographs. Therefore, we have decided to "clean" the output by removing the edges that violate the mathematically defined orthology relation. The results can be found in Tables 8.2-8.5. OrthoMCLean refers to the later described datasets.

Method	$\|V\|$	$\|E\|$	$\|CC\|$	$\|smallCC\|$	$\|K_4\|$	$\|K_3\|$	$\|K_2\|$	%Forbs	$\%P_4$-s	%Cographs	%s_CC
BBH	668	5228	29	3	0	1	2	2.2	1.2	85.9	10.7
Proteinortho	772	5588	72	28	0	4	12	6.0	1.5	52.0	40.5
PoFF	731	4594	92	52	0	5	37	1.3	1.0	42.2	55.5
OrthoMCL	1563	151682	23	1	0	1	0	2.1	1.0	91.5	5.4
OrthoMCLean	1258	115817	21	1	0	1	0	0.9	0.6	93.5	5.0

Table 8.2: Results form the four methods when taking 1000 samples of induced subgraphs in five vertices for the dataset with 20 species and 20 gene families. In the `orthoMCL` graph, 23.64% of the edges connect pairs of genes of the same species.

Method	$\|V\|$	$\|E\|$	$\|CC\|$	$\|smallCC\|$	$\|K_4\|$	$\|K_3\|$	$\|K_2\|$	%Forbs	%P_4-s	%Cographs	%s_CC
BBH	3797	15741	557	229	6	16	203	1.2	0.8	57.2	40.8
Proteinortho	4052	19216	582	217	6	20	162	1.7	1.2	58.9	38.2
PoFF	3758	14777	556	151	2	14	94	3.0	1.7	66.8	28.5
OrthoMCL	8280	883823	269	67	1	4	61	1.9	1.2	72.7	24.2
OrthoMCLean	4716	363469	249	63	1	3	58	1.0	0.5	72.5	26.0

Table 8.3: Results of the tools when taking 1000 samples of induced subgraphs in five vertices for the dataset with 20 species and 50 gene families. In the `orthoMCL` graph, 58.87% of the edges connect pairs of genes of the same species.

Method	$\|V\|$	$\|E\|$	$\|CC\|$	$\|smallCC\|$	$\|K_4\|$	$\|K_3\|$	$\|K_2\|$	%Forbs	$\%P_4$-s	%Cographs	%s_CC
BBH	5204	22157	827	546	142	224	129	3.4	0.2	30.5	65.9
Proteinortho	5783	26791	794	397	116	95	88	5.7	1.2	42.8	50.3
PoFF	5007	20077	873	551	9	23	190	4.6	0	32.4	61.6
OrthoMCL	15038	2583597	417	195	60	66	51	2.9	1.5	50.5	45.1
OrthoMCLean	6482	453554	383	189	59	63	44	2.6	0.4	47.1	49.9

Table 8.4: Results of the tools when taking 1000 samples of induced subgraphs in five vertices for the dataset with 20 species and 80 gene families. In the `orthoMCL` graph, 82.44% of the edges connect pairs of genes of the same species.

Method	$\|V\|$	$\|E\|$	$\|CC\|$	$\|smallCC\|$	$\|K_4\|$	$\|K_3\|$	$\|K_2\|$	%Forbs	$\%P_4$-s	%Cographs	%s_CC
BBH	14430	49312	2656	1549	13	485	1001	2.0	0.3	39.1	58.6
Proteinortho	15718	173909	2203	1036	13	374	548	4.2	0.6	49.1	46.1
PoFF	15350	43946	2951	1578	6	295	792	3.3	2.3	38.4	56

Table 8.5: Results of the tools when taking 1000 samples of induced subgraphs in five vertices for the dataset with 20 species and 100 gene families. `orthoMCL` could not be applied to this dataset since it would have taken more than one month for the calculation of the graph.

As we can obserbve the BBH tends to create many more small clusters than the other methods. Proteinortho, PoFF and OrthoMCL perform similarly, althought Proteinortho and PoFF create more small clusters than OrthoMCL, this can be explained by the graph-construction phase in the former tools, since they make use of the BBH method and then will try to add paralagous genes to the cluster if the putative paralogs have at least a 95% of sequence similarity to the BBH gene, while OrthoMCL has no restrictions before the clustering phase. For all the tools, the percentage of forbidden subgraphs is very low and therefore the graphs can be edited to convert them to cographs.

We have also run Proteinortho and PoFF on the set of 12 metazon proteomes and obtained the corresponding orthology graphs. The aim here was to compare whether PoFF, the extended version of Proteinortho which accounts for synteny information, however we can observe that they perfom pretty similar, with the exception that PoFF created more small clusters than Proteinortho. The results can be found in the Table 8.6.

Method	$\|V\|$	$\|E\|$	$\|CC\|$	$\|smallCC\|$	$\|K_4\|$	$\|K_3\|$	$\|K_2\|$	%Forbs	$\%P_4$-s	%Cographs	%s_CC
Proteinortho	189620	363351	32981	16242	5	157	6960	5.52	2.77	42.23	49.48
PoFF	165760	176436	39524	24988	10	249	10481	8.22	3.89	25.23	62.66

Table 8.6: Results for `Proteinortho` and `PoFF` in the set of 12 metazon proteomes. We took 10 000 samples of induced subgraphs in five vertices.

8.4 Concluding Remarks

We propose an algorithm that simulates gene family histories akin to real data. This will allow reconstruction algorithms to measure their accuracy and performance. Given a certain reconstruction method one might ask if the orthology matrix could be deduced from the inferred reconciled tree or if the homology relation between the genes was predicted correctly. Furthermore it could be analysed if the method was able to infer the gene duplications and losses. A method that is able to detect large scale duplications will then identify the cluster and genome duplications generated by our algorithm.

We have also presented a method to measure "noise" in orthology relations based on induced subgraphs on five vertices that are divided in four categories: forbidden subgraphs, P_4-sparse, cographs and small connected components. Depending on the percentage of these subgraphs found in the corresponding orhtology relation graph from a dataset, we are able to say whether the graphs is closer to a cograph or to a random graph. If a graph contains less than 20% of forbidden subgraphs, it is likely that it can be edited and therefore converted to a cograph.

We have applied this approach to the output of four methods to measure their performance when tested with simulated data: `BBH`, `Proteinortho`, `PoFF` and `OrthoMCL`. Surprisingly, we found out that the graph obtained from the `OrthoMCL` output contains edges that connect pairs of genes from the same species, which violates the definition of orthology.

We have also tested the datasets found in the `OMA` database and found that the orthology graph induced by pairs of orthologs has few errors and therefore one should be interested in finding out which of those pairs of orthologs are false positives to obtain a perfect valid ortholoy relation. We also applied the method to the output graphs from `Proteinortho` and `PoFF` when applied to a set of 12 metazoan proteoms. We can conclude that both methods perform very well and therefore create orthology relation graphs that are close to cographs, therefore it might be interesting to think of a heuristic that takes two lists of edges: one with strong evidence and one with weak evidence, such that the edges in the former should not be removed when applying a cograph editing method and edges in the second list can be the first ones to be taken into account for removal.

CHAPTER 9

Conclusions

Orthology refers specifically to the relationship between two genes that arose by a speciation event, recent or remote. Comparing orthologous genes is essential to the correct reconstruction of species trees, so that detecting and identifying orthologous genes is an important problem, and a longstanding challenge, in comparative and evolutionary genomics and phylogenetics. In this work we were concerned about answering the following question: *How much information about the gene tree, the species tree, and their reconciliation is already contained in the orthology relation among genes?*

In this thesis we have presented a new characterization for orthology relations based on previous results on symbolic ultrametrics. As a result we have found out that valid orthology relations can be characterized as a well-studied class of graphs called cographs, which is precisely the class of graphs that do not contain induced paths on any subset of four vertices. A cograph has a unique associated cotree, whose internal nodes have labels 1 or 0. In the case of phylogenetic trees, these correspond to "speciation" and "duplication" events, respectively. Usually this tree will not be completely resolved, but can be used as a constraint to resolve multifurcating nodes that represent contractions of several events of the same type.

We have shown that cliques in a certain graph that can be associated to a symbolic ultrametric are closely related to the structure of its discriminating symbolic representation which turns out to be a cotree. We use this result to help derive a new algorithm for determining whether a map is a symbolic ultrametric or not. The algorithm is called `BOTTOM-UP` and is based on cliques, cherries and partitions for recovering phylogenetic trees. It would be interesting to find an extension of this algorithm, like the `MIN-CUT supertree` algorithm which extends the `BUILD` algorithm in such a way that, when one of the conditions is not satisfied, the algorithm can continue by applying some sort of `MIN-CUT` or a

more suitable approach.

With the characterization of event-labeled gene trees we have shown how to derive a species tree by reducing the reconciliation map from a gene tree to a species tree to rooted triples of genes residing in three distinct species. We have proved that the set of induced triples from the gene tree whose root is a speciation node are the ones that define the structure of the corresponding species tree. Furthermore, we have shown that event labeled gene trees for which a species tree exists can be characterized in terms of set of triples of that type. Simulated data shows, moreover, that such trees convey a large amount of information on the underlying species tree.

As described in Chapter 2 it has been shown that any graph has an associated modular decomposition tree whose internal nodes represent series, parallel and prime modules. Series and parallel modules correspond to labels 1 and 0 in a cotree, respectively. In general a tree structure is simpler than a graph structure, therefore it would be interesting to find heuristics to edit a graph to a cograph by starting resolving prime nodes in the corresponding modular decomposition tree.

It would be interesting to investigate graphs with weighted edges such that edges with strong evidence about the orthology are more likely to be kept than edges weak edges. One could think of using a method like the graph completion problem or cograph editing that additionally takes into account such weights. Moreover, one should be interested in the investigation of different approaches for the cograph editing problem when applied to the same graph, in such a way that each outputed cograph has a weight and the space of cographs can be analyzed to find the best solution. Furthermore, for the inference of species trees, finding the maximum set of consistent triples, such that each triple has a weight that represents evidence for orthology could help in the accuracy of the species tree when dealing with real data.

Alternatively, given an arbitrary (weighted) orthology map it may be possible to find the closest cograph in terms of edge editing operations, using Integer Linear Programming (ILP) approaches. Therefore a binary variable for each possible edge is created. A set of constraints is defined forbidding each possible induced path of length four and an objective function minimizing the sum of (weighted) edge editing operations. However, for the resulting cograph a species tree may not always exist. To assure its existence the rooted species triples extracted from the cograph have to be compatible. This can be achieved using the ideas from Chang et al. [2011] by forcing the existence of a taxon-cluster representation for the species tree compatible with all the species triples. This cluster representation of a tree is a hierarchical order of subsets of the leaf set. A node from the species tree is described by the set of its descendant leaves. Therefore the corresponding leaf sets of two nodes have to be either distinct or one set is included into the other, which can be easily checked within ILP using the three-gamete condition [Gusfield, 1997].

It has been important to find a method to measure “noise” in orthology graphs which would enable us to judge whether this graph is closer to a cograph or to random one. We have developped a method based on “forbidden subgraphs”, induced subgraphs on five vertices that contain more than one P_4, P_4-sparse graps, graphs which contain only one P_4, and cographs, as a benchmark for orthology detection

methods. We have applied our approach to output graphs produced by BBH, Proteinortho, PoFF and OrthoMCL to test the accuracy of their predictions. We have also tested real data sets found in the OMA database as well as the predictions obtained from Proteinortho and PoFF when applied to a set of 12 Metazoan proteomes. The results show that the predictions of the tools are quite accurate and that those graphs can be edited to be converted to a cograph which in turn will represent a valid orthology relation. However, why orthoMCL predicts orthology relations between pairs of genes that belong to the same species still remains an open question.

It is worth mentioning that the general theory developed here for duplications and speciations is potentially useful for more refined applications. More specifically, gene duplications have several different mechanistic causes that are also empirically distinguishable in real data sets. Thus it could be of interest, for example, to consider data sets which, as well as representing speciation and duplication events could also take into account events such as local segmental duplications, duplications by retrotransposition, or whole-genome duplications [Zhang, 2003]. Moreover, in addition to such events, it might be of interest to consider lineage sorting and horizontal gene transfer, both of which play an important role in genome evolution [Maddison, 1997, Page and Charleston, 1998]. From the point of view of gene trees, these behave in a similar manner to speciations, although they introduce incongruencies between the gene and species trees. Hence it might be of interest to investigate whether some of the theory developed in this work could be extended to phylogenetic networks which are graph theoretical structures generalizing phylogenetic trees which are commonly used for modeling horizontal gene transfer (see e.g. [Huson et al., 2010]).

List of Figures

List of Tables

Bibliography

Sebastian Böcker and Andreas W. M. Dress. Recovering symbolically dated, rooted trees from symbolic ultrametrics. *Adv. Math.*, 138:105–125, 1998.

Charles Semple and Mike Steel. *Phylogenetics*, volume 24 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, UK, 2003.

Walter M. Fitch. Homology: a personal view on some of the problems. *Trends Genet.*, 16:227–231, 2000.

D. G. Corneil, H. Lerchs, and L K Stewart Burlingham. Complement reducible graphs. *Discr. Appl. Math.*, 3:163–174, 1981.

Cedric Chauve and Nadia El-Mabrouk. New perspectives on gene family evolution: Losses in reconciliation and a link with supertrees. *LNCS*, 5541:46–58, 2009.

Marcus Lechner, Sven Findeiß, Lydia Steiner, Manja Marz, Peter F. Stadler, and Sonja J. Prohaska. `Proteinortho:` detection of (co-)orthologs in large-scale analysis. *BMC Bioinformatics*, 12:124, 2011.

L. Li, C. J. Stoeckert, and D. S. Roos. OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res.*, 13(9):2178–2189, Sep 2003.

Andreas Brandstädt, Van Bang Le, and Jeremy P Spinrad. *Graph Classes: A Survey*. SIAM Monographs on Discrete Mathematics and Applications. Soc. Ind. Appl. Math., Philadephia, 1999.

D. G. Corneil, Y. Perl, and L K Stewart Burlingham. A linear recognition algorithm for cographs. *SIAM J. Computing*, 14:926–934, 1985.

Hans-Jürgen Bandelt. Recognition of Tree Metrics. *SIAM J. Discrete Math.*, 3(1):1–6, February 1990.

Walter M. Fitch. Distinguishing homologous from analogous proteins. *Syst. Zool.*, 19:99–113, 1970.

G. Gray and Walter M. Fitch. Evolution of antibiotic resistance genes: The dna sequence of a kanamycin resistance gene from staphylococcus aureus. *Mol. Biol. Evol.*, 1:57–66, 1983.

David M. Kristensen, Yuri I. Wolf, Arcady R. Mushegian, and Eugene V. Koonin. Computational methods for gene orthology inference. *Briefings Bioinf.*, 2011. doi:10.1093/bib/bbr030.

K. Dittmar and D. Liberles. *Evolution after Gene Duplication*. Wiley-Blackwell, 2010.

M. T. Hallett and J. Lagergren. New algorithms for the duplication-loss model. In *Proceedings of the fourth annual international conference on Computational molecular biology*, RECOMB '00, pages 138–146, New York, NY, USA, 2000. ACM. ISBN 1-58113-186-0. doi: 10.1145/332306.332359. URL `http://doi.acm.org/10.1145/332306.332359`.

C. E. Storm and E. L. Sonnhammer. Automated ortholog inference from phylogenetic trees and calculation of orthology reliability. *Bioinformatics*, 18(1):92–99, Jan 2002.

James S. Farris. Estimating Phylogenetic Trees from Distance Matrices. *The American Naturalist*, 106(951):645–668, 1972.

C. M. Zmasek and S. R. Eddy. RIO: analyzing proteomes by automated phylogenomics using resampled inference of orthologs. *BMC Bioinformatics*, 3:14, May 2002.

Jaime Huerta-Cepas, Hernan Dopazo, Joaquin Dopazo, and Toni Gabaldon. The human phylome. *Genome Biology*, 8:R109, 2007.

H Li, A Coghlan, J Ruan, L J Coin, J K Hériché, L Osmotherly, R Li, T Liu, Z Zhang, L Bolund, G K Wong, W Zheng, P Dehal, J Wang, and R Durbin. `TreeFam`: a curated database of phylogenetic trees of animal gene families. *Nucleic Acids Res.*, 34:D572–D580, 2006.

A. J. Vilella, J. Severin, A. Ureta-Vidal, L. Heng, R. Durbin, and E. Birney. EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates. *Genome Res.*, 19(2):327–335, Feb 2009.

Leszek P. Pryszcz, Jaime Huerta-Cepas, and Toni Gabaldon. `MetaPhOrs`: orthology and paralogy predictions from multiple phylogenetic evidence using a consistency-based confidence score. *Nucleic Acids Res.*, 39:e32, 2011.

R. T. van der Heijden, B. Snel, V. van Noort, and M. A. Huynen. Orthology prediction at scalable resolution by phylogenetic tree analysis. *BMC Bioinformatics*, 8:83, 2007.

M. S. Poptsova and J. P. Gogarten. BranchClust: a phylogenetic algorithm for selecting gene families. *BMC Bioinformatics*, 8:120, 2007.

R. Jothi, E. Zotenko, A. Tasneem, and T. M. Przytycka. COCO-CL: hierarchical clustering of homology relations based on evolutionary correlations. *Bioinformatics*, 22(7):779–788, Apr 2006.

J. F. Dufayard, L. Duret, S. Penel, M. Gouy, F. Rechenmann, and G. Perriere. Tree pattern matching in phylogenetic trees: automatic search for orthologs or paralogs in homologous gene sequence databases. *Bioinformatics*, 21(11):2596–2603, Jun 2005.

A. C. Berglund-Sonnhammer, P. Steffansson, M. J. Betts, and D. A. Liberles. Optimal gene trees from sequences and species trees using a soft interpretation of parsimony. *J. Mol. Evol.*, 63(2): 240–250, Aug 2006.

Rouchira S. Datta, Christopher Meacham, Bushra Samad, Christoph Neyer, and Kimmen Sjölander. Berkeley PHOG: Phylofacts orthology group prediction web server. *Nucl. Acids Res.*, 37:W84–W89, 2009.

Joseph Felsenstein. *Inferring Phylogenies*. Sunderland MA: Sinauer Associates, 2004.

P. Puigbo, Y. I. Wolf, and E. V. Koonin. The tree and net components of prokaryote evolution. *Genome Biol Evol*, 2:745–756, 2010.

G. Fang, N. Bhardwaj, R. Robilotto, and M. B. Gerstein. Getting started in gene orthology and functional analysis. *PLoS Comput. Biol.*, 6(3):e1000703, Mar 2010.

M Remm, C E Storm, and E L Sonnhammer. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J. Mol. Biol.*, 314:1041–1052, 2001.

B. Linard, J. D. Thompson, O. Poch, and O. Lecompte. OrthoInspector: comprehensive orthology analysis and visual exploration. *BMC Bioinformatics*, 12:11, 2011.

T. W. Chen, T. H. Wu, W. V. Ng, and W. C. Lin. DODO: an efficient orthologous genes assignment tool based on domain architectures. Domain based ortholog detection. *BMC Bioinformatics*, 11 Suppl 7:S6, 2010.

D. P. Wall, H. B. Fraser, and A. E. Hirsh. Detecting putative orthologs. *Bioinformatics*, 19(13): 1710–1711, Sep 2003.

L. B. Koski and G. B. Golding. The closest BLAST hit is often not the nearest neighbor. *Journal of molecular evolution In Journal of Molecular Evolution*, 52:540–542, June 2001.

Roman L. Tatusov, Eugene V. Koonin, and David J. Lipman. A genomic perspective on protein families. *Science*, 278(5338):631–637, October 1997.

A. M. Altenhoff, A. Schneider, G. H. Gonnet, and C. Dessimoz. OMA 2011: orthology inference among 1000 complete genomes. *Nucleic Acids Res.*, 39(Database issue):D289–294, Jan 2011.

A. C. Roth, G. H. Gonnet, and C. Dessimoz. Algorithm of OMA for large-scale orthology inference. *BMC Bioinformatics*, 9:518, 2008.

L. J. Jensen, P. Julien, M. Kuhn, C. von Mering, J. Muller, T. Doerks, and P. Bork. eggNOG: automated construction and annotation of orthologous groups of genes. *Nucleic Acids Res.*, 36 (Database issue):D250–254, Jan 2008.

R. M. Waterhouse, F. Tegenfeldt, J. Li, E. M. Zdobnov, and E. V. Kriventseva. OrthoDB: a hierarchical catalog of animal, fungal and bacterial orthologs. *Nucleic Acids Res.*, 41(Database issue):D358–365, Jan 2013.

Stijn van Dongen. *Graph clustering by Flow Simulation*. PhD Thesis, University of Utrecht, Cambridge, UK, 2000.

J. Jun, I. I. Mandoiu, and C. E. Nelson. Identification of mammalian orthologs using local synteny. *BMC Genomics*, 10:630, 2009.

S. B. Cannon and N. D. Young. OrthoParaMap: distinguishing orthologs from paralogs by integrating comparative genome data and gene phylogenies. *BMC Bioinformatics*, 4:35, Sep 2003.

L. Goodstadt and C. P. Ponting. Phylogenetic reconstruction of orthology, paralogy, and conserved synteny for dog and human. *PLoS Comput. Biol.*, 2(9):e133, Sep 2006.

I. Wapinski, A. Pfeffer, N. Friedman, and A. Regev. Automatic genome-wide reconstruction of phylogenetic gene trees. *Bioinformatics*, 23:i549–558, Jul 2007.

G. Shi, L. Zhang, and T. Jiang. MSOAR 2.0: Incorporating tandem duplications into ortholog assignment based on genome rearrangement. *BMC Bioinformatics*, 11:10, 2010.

K. Mahmood, A. S. Konagurthu, J. Song, A. M. Buckle, G. I. Webb, and J. C. Whisstock. EGM: encapsulated gene-by-gene matching to identify gene orthologs and homologous segments in genomes. *Bioinformatics*, 26(17):2076–2084, Sep 2010.

David Sankoff. *OMG! Orthologs for Multiple Genomes - Competing Formulations*, volume 6674 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-21259-8.

O. R. Bininda-Emonds. The evolution of supertrees. *Trends Ecol. Evol. (Amst.)*, 19(6):315–322, Jun 2004a.

Monika Rauch Henzinger, Valerie King, and Tandy Warnow. Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology. *Algorithmica*, 24:1–13, 1999.

T. Jiang, Y. Xu, and M. Zhang. *Current Topics in Computational Molecular Biology*. MIT Press, 2002.

M. A. Ragan. Phylogenetic inference based on matrix representation of trees. *Mol. Phylogenet. Evol.*, 1(1):53–58, Mar 1992.

Bernard R. Baum. Combining Trees as a Way of Combining Data Sets for Phylogenetic Inference, and the Desirability of Combining Gene Trees. *Taxon*, 41(1):3–10, 1992. ISSN 00400262. doi: 10.2307/1222480. URL `http://dx.doi.org/10.2307/1222480`.

Francois-Joseph Lapointe and Guy Cucumel. The average consensus procedure: Combination of weighted trees containing identical or overlapping sets of taxa. *Systematic Biology*, 46(2):306–312, 1997. doi: 10.1093/sysbio/46.2.306. URL `http://sysbio.oxfordjournals.org/content/46/2/306.abstract`.

O. Eulenstein, D. Chen, J. G. Burleigh, D. Fernandez-Baca, and M. J. Sanderson. Performance of flip supertree construction with a heuristic algorithm. *Syst. Biol.*, 53(2):299–308, Apr 2004.

Andy Purvis. A Modification to Baum and Ragan's Method for Combining Phylogenetic Trees. *Systematic Biology*, 44(2):251–255, 1995. ISSN 10635157. doi: 10.2307/2413710. URL `http://dx.doi.org/10.2307/2413710`.

O.R.P Bininda-Emonds. *Phylogenetic Supertrees*. Kluwer Academic Press, Dordrecht, The Netherlands, 2004b.

A. V. Aho, Y. Sagiv, T. G. Szymanski, and J. D. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM J. Comput.*, 10: 405–421, 1981a.

A. V. Aho, Y. Sagiv, T. G. Szymanski, and J. D. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM J. Comput.*, 10: 405–421, 1981b.

Jesper Jansson, Richard S. Lemence, and Andrzej Lingas. The complexity of inferring a minimally resolved phylogenetic supertree. *SIAM J. Comput.*, 41:272–291, 2012.

Meei Pyng Ng and Nicholas C. Wormald. Reconstruction of rooted trees from subtrees. *Discr. Appl. Math.*, 69:19–31, 1996.

D. Bryant and M. Steel. Extension operations on sets of leaf-labeled trees. *Adv. Appl. Math.*, 16: 425–453, 1995.

Stefan Grünewald, Mike Steel, and M. Shel Swenson. Closure operations in phylogenetics. *Math. Biosci.*, 208:521???537, 2007.

Leszek Gasieniec, Jesper Jansson, Andrzej Lingas, and Anna Östlin. On the complexity of constructing evolutionary trees. *Journal of Combinatorial Optimization*, 3:183–197, 1999.

Jesper Jansson. On the complexity of inferring rooted evolutionary trees. *Electronic Notes Discr. Math.*, 7:50–53, 2001.

Y. J. He, T. N. Huynh, J. Jansson, and W. K. Sung. Inferring phylogenetic relationships avoiding forbidden rooted triplets. *J Bioinform Comput Biol*, 4:59–74, 2006.

J. Byrka, S. Guillemot, and J. Jansson. New results on optimizing rooted triplets consistency. *Discr. Appl. Math.*, 158:1136–1147, 2010a.

Bang Ye Wu. Constructing the maximum consensus tree from rooted triples. *J. Comb. Optimization*, 8:29–39, 2004.

J. Byrka, P. Gawrychowski, K. T. Huber, and S. Kelk. Worst-case optimal approximation algorithms for maximizing triplet consistency within phylogenetic networks. *J. Discr. Alg.*, 8:65–75, 2010b.

C. Semple and M. Steel. A supertree method for rooted trees. *Discrete Applied Mathematics*, 105: 147–158, 2000.

Roderic D. M. Page. Modified mincut supertrees. In *Proceedings of the Second International Workshop on Algorithms in Bioinformatics*, WABI '02, pages 537–552, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-44211-1. URL `http://dl.acm.org/citation.cfm?id=645907.673126`.

S. Snir and S. Rao. Using max cut to enhance rooted trees consistency. *IEEE/ACM Trans Comput Biol Bioinform*, 3(4):323–333, 2006.

Charles Semple. Reconstructing minimal rooted trees. *Discr. Appl. Math*, 127:489–503, 2003.

A M Altenhoff and C. Dessimoz. Phylogenetic and functional assessment of orthologs inference projects and methods. *PLoS Comput Biol.*, 5:e1000262, 2009.

C. Falls, B. Powell, and J. Snœyink. Computing high-stringency COGs using Turán-type graphs. Technical report, http://www.cs.unc.edu/∼snoeyink/comp145/cogs.pdf, 2008.

R L Tatusov, M Y Galperin, D A Natale, and E V Koonin. The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Res*, 28:33–36, 2000.

A C Berglund, E Sjölund, G Ostlund, and E L Sonnhammer. `InParanoid 6`: eukaryotic ortholog clusters with inparalogs. *Nucleic Acids Res.*, 36:D263–D266, 2008.

D L Wheeler, T Barrett, D A Benson, S H Bryant, K Canese, V Chetvernin, D M Church, M Dicuccio, R Edgar, S Federhen, M Feolo, L Y Geer, W Helmberg, Y Kapustin, O Khovayko, D Landsman, D J Lipman, T L Madden, D R Maglott, V Miller, J Ostell, K D Pruitt, G D Schuler, M Shumway, E Sequeira, S T Sherry, K Sirotkin, A Souvorov, G Starchenko, R L Tatusov, T A Tatusova, L Wagner, and E Yaschenko. Database resources of the national center for biotechnology information. *Nucleic Acids Res.*, 36:D13–D21, 2008.

Yunlong Liu, Jianxin Wang, Jiong Guo, and Jianer Chen. Cographs editing: Complexity and parametrized algorithms. In B. Fu and D. Z. Du, editors, *COCOON 2011*, volume 6842 of *Lect. Notes Comp. Sci.*, pages 110–121, Berlin, Heidelberg, 2011. Springer-Verlag.

Fábio Protti, Maise Dantas da Silva, and Jayme Luiz Szwarcfiter. Applying modular decomposition to parameterized cluster editing problems. *Th. Computing Syst.*, 44:91–104, 2009.

Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, September 1973. ISSN 0001-0782. doi: 10.1145/362342.362367. URL `http://doi.acm.org/10.1145/362342.362367`.

John D. Eblen, Charles A. Phillips, Gary L. Rogers, and Michael A. Langston. The maximum clique enumeration problem: algorithms, applications and implementations. In *Proceedings of the 7th international conference on Bioinformatics research and applications*, ISBRA'11, pages 306–319, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-21259-8. URL `http://dl.acm.org/citation.cfm?id=2009164.2009194`.

Makino Kazuhisa and Uno Takeaki. New algorithms for enumerating all maximal cliques. pages 260–272. Springer-Verlag, 2004.

Matthew C. Schmidt, Nagiza F. Samatova, Kevin Thomas, and Byung-Hoon Park. A scalable, parallel algorithm for maximal clique enumeration. *J. Parallel Distrib. Comput.*, 69(4):417–428, April

2009. ISSN 0743-7315. doi: 10.1016/j.jpdc.2009.01.003. URL http://dx.doi.org/10.1016/j.jpdc.2009.01.003.

P. Sneath and R. Sokal. *Numerical Taxonomy*. W.H. Freeman and Company, San Francisco, 1973. pp. 230–234.

Maribel Hernandez-Rosales, Marc Hellmuth, Nicolas Wieseke, Katharina Huber, Vincent Moulton, and Peter Stadler. From event-labeled gene trees to species trees. *BMC Bioinformatics*, 13(Suppl 19):S6, 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-S19-S6. URL http://www.biomedcentral.com/1471-2105/13/S19/S6.

Marc Hellmuth, Maribel Hernandez-Rosales, Katharina T. Huber, Vincent Moulton, Peter F. Stadler, and Nicolas Wieseke. Orthology relations, symbolic ultrametrics, and cographs. *Journal of Mathematical Biology*, 66(1-2):399–420, 2013. doi: 10.1007/s00285-012-0525-x. URL http://dx.doi.org/10.1007/s00285-012-0525-x.

Jean-Philippe Doyon, Cedric Chauve, and Sylvie Hamel. Space of gene/species trees reconciliations and parsimonious models. *J. Comp. Biol.*, 16:1399–1418, 2009.

L. Zhang. On a Mirkin-Muchnik-Smith conjecture for comparing molecular phylogenies. *J. Comput. Biol.*, 4:177–187, 1997.

Mariana Constantinescu and David Sankoff. An efficient algorithm for supertrees. *J Classification*, 12:101–112, 1995.

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

Paola Bonizzoni, Gianluca Della Vedova, and Riccardo Dondi. Reconciling a gene tree to a species tree under the duplication cost model. *Theor. Comp. Sci.*, 347:36–53, 2005.

J. G. Burleigh, M. S. Bansal, A. Wehe, and O. Eulenstein. Locating large-scale gene duplication events through reconciled trees: implications for identifying ancient polyploidy events in plants. *J. Comput. Biol.*, 16:1071–1083, 2009.

P. Górecki and Tiuryn J. DSL-trees: A model of evolutionary scenarios. *Theor. Comp. Sci.*, 359: 378–399, 2006.

M. Goodman, Czelusniak J., 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. *Systematic Zoology*, 28:132–163, Jun 1979.

S. Keller-Schmidt, M. Tuğrul, V. M. Eguíluz, E. Hernández-Garcíi, and K. Klemm. An age dependent branching model for macroevolution. Technical Report 1012.3298v1, arXiv, 2010.

S. Ohno. Gene duplication and the uniqueness of vertebrate genomes circa 1970-1999. *Semin. Cell Dev. Biol.*, 10:517–522, Oct 1999.

M. Lynch and J. S. Conery. The evolutionary fate and consequences of duplicate genes. *Science*, 290: 1151–1155, 2000.

S. J. Prohaska, C. Fried, C. Flamm, G. P. Wagner, and P. F. Stadler. Surveying phylogenetic footprints in large gene clusters: applications to Hox cluster duplications. *Mol. Phylogenet. Evol.*, 31:581–604, May 2004.

W Xu, C Zheng, and D Sankoff. Paths and cycles in breakpoint graph of random multichromosomal genomes. *J Comput Biol*, 14(4):423–435, May 2007.

B. Jamison and S. Olariu. Recognizing p_4-sparse graphs in linear time. *SIAM Journal on Computing*, 21(2):381–406, 1992. doi: 10.1137/0221027. URL `http://epubs.siam.org/doi/abs/10.1137/0221027`.

C. Hoang. Perfect graphs. *PhD thesis, McGill University*, 1985.

Sarah Berkemer. Cograph Editing: An Approach to Adjust the Orthology Relation for the Reconstruction of Phylogenetic Trees. *Bachelor Thesis, Saarland University*, 2012.

Michel Habib, Fabien Montgolfier, and Christophe Paul. A simple linear-time modular decomposition algorithm for graphs, using order extension. In Torben Hagerup and Jyrki Katajainen, editors, *Algorithm Theory - SWAT 2004*, volume 3111 of *Lecture Notes in Computer Science*, pages 187–198. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-22339-9. doi: 10.1007/978-3-540-27810-8_17. URL `http://dx.doi.org/10.1007/978-3-540-27810-8_17`.

Marcus Lechner, Maribel Hernandez-Rosales, Daniel Doerr, Nicolas Wieseke, Annelyse Thevenin, Jens Stoye, Sonja J. Prohaska, and Peter F. Stadler. Orthology detection combining clustering and synteny for very large data sets. *Submitted*, 2013.

Daniel Doerr, Annelyse Thévenin, and Jens Stoye. Gene family assignment-free comparative genomics. *BMC Bioinformatics*, 13:S3 19, 2012.

C L Strope, K Abel, S D Scott, and E N Moriyama. Biological sequence simulation for testing complex evolutionary hypotheses: indel-Seq-Gen version 2.0. *Mol. Biol. Evol.*, 26:2581–2593, 2009.

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

Dan Gusfield. *Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology*. Cambridge University Press, 1997. ISBN 0-521-58519-8.

Jianzhi Zhang. Evolution by gene duplication: an update. *Trends Ecol. Evol.*, 18:292–298, 2003.

Wayne P. Maddison. Gene trees in species trees. *Syst. Biol.*, 46:523–536, 1997.

Roderic D. M. Page and Michael A. Charleston. Trees within trees: phylogeny and historical associations. *Trends Ecol. Evol.*, 13:356–359, 1998.

D. Huson, R. Rupp, and C. Scornavacca. *Phylogenetic Networks*. Cambridge University Press, 2010.

Curriculum Scientiae

EDUCATION:

since 03/2008	PhD student at University of Leipzig • Group of Prof. Peter F. Stadler, Chair of Bioinformatics • Thesis: *The Orthology Road, Theory and Methods in Orthology Analysis.*
2007 – 2008	Master Degree Project, University of Leipzig, Germany. • Project: *Identification of ncRNA predictions and their targets using bioinformatics tools and biological information.*
2004 – 2007	Master Degree studies of Bioinformatics at University of Montreal, Canada • Project 1: *No evidence for relation between Co-regulation and Regulatory Interactions in the evolution of Regulatory Networks* • Project 2: *Identification of motifs in microRNAs and their target sites.*
1996 – 2000	Bachelor Degree studies in Computer Science at Universidad Nacional Autónoma de México (UNAM) in Mexico City, Mexico • Thesis: *Application of a 3D chain code for the one-dimensional representation of digital images.*

WORKING EXPERIENCE:

2002 – 2004	Software Engineer at the Genomics Research Center (CCG), UNAM, Cuernavaca, Mexico

	• Group of Prof. Julio Collado-Vides, Department of Computational Genomics • Project 1: *annotation of the complete genome of Nitrogen Fixing Bacteria-Rhizobium Etli.* • Project 2: *Analysis and design of the navigation of the project Multigenome Web Site.* • Project 3: *Data Analysis for the Regulation Network of E. coli project.* • Project 4: *Operon interaction network based on gene ontologies and the regulatory information from regulonDB.*
2000 – 2001	Programmer at Aranea S.A. de C.V., Mexico City, Mexico.
1999 – 2000	Programmer at Alterbase S.A. de C.V., Mexico City, Mexico.
1998 – 2000	Teaching Assistent at the Faculty of Science, UNAM, Mexico City, Mexico.

IT-KNOWLEDGE:

OPERATING SYSTEMS:	UNIX, Mac, Linux, Windows
PROGRAMMING:	C, C++, Perl, R, Prolog, Lisp, Delphi.
MARKUP LANGUAGES:	Latex
DATABASE SYSTEMS:	MySQL, Oracle

LANGUAGE SKILLS:

SPANISH:	native speaker
ENGLISH:	fluent
FRENCH:	fluent
GERMAN:	intermediate

Publications

JOURNALS:

Marcus Lechner, Maribel Hernandez-Rosales, Daniel Doerr, Nicolas Wieseke, Annelyse Thevenin, Jens Stoye, Sonja J. Prohaska and Peter F. Stadler. (2013).
Orthology Detection Combining Clustering and Synteny for Very Large Data Sets. *Submitted*.

Marc Hellmuth, Maribel Hernandez-Rosales, Katharina T. Huber, Vincent Moulton, Peter F. Stadler, and Nicolas Wieseke. (2013).
Orthology relations, symbolic ultrametrics, and cographs. *J. Math. Biol.* 66(1-2):399-420.

Maribel Hernandez-Rosales, Marc Hellmuth, Nicolas Wieseke, Katharina Huber, Vincent Moulton, and Peter F. Stadler. (2012).
From event-labeled gene trees to species trees. *BMC Bioinformatics* 13(Suppl. 19):S6.

Ulrike Mueckstein, Hakim Tafer, Stephan H. Bernhart Maribel Hernandez-Rosales, Jorg Vogel, Peter F. Stadler, Ivo L.Hofacke. (2008).
Translational Control by RNA-RNA Interaction: Improved Computation of RNA-RNA Binding Thermodynamics. *BioInformatics Research and Development (BIRD)* 13:114-127.

CONFERENCES / SEMINARS:

Max Planck Institute for Mathematics in the Sciences Seminar (Presenter)
Hernandez-Rosales M, Marc Hellmuth, Nicolas Wieseke, Katharina Huber, Vincent Moulton, and Peter F. Stadler.: *From Orthology Relations, through Cographs and Event-labeled Gene Trees to Spceies Trees*
26/02/2013; Leipzig, Germany

Insitute of Exact Mathematics, Faculty of Computer Science, University of Brasilia, Seminar (Presenter)
Hernandez-Rosales M, Marc Hellmuth, Nicolas Wieseke, Katharina Huber, Vincent Moulton, and Peter F. Stadler.: *From Orthology Relations, through Cographs and Event-labeled Gene Trees to Spceies Trees*
05/11/2012; Brasilia, Brazil

University of Fiocruz, Seminar (Presenter)
Hernandez-Rosales M, Marc Hellmuth, Nicolas Wieseke, Katharina Huber, Vincent Moulton, and Peter F. Stadler.: *From Orthology Relations, through Cographs and Event-labeled Gene Trees to Spceies Trees*
24/10/2012; Rio de Janeiro, Brazil

RECOMB-CG 2012 (Presenter)
Hernandez-Rosales M, Marc Hellmuth, Nicolas Wieseke, Katharina Huber, Vincent Moulton, and Peter F. Stadler.: *From Event-labeled Gene Trees to Spceies Trees*
24/10/2012; Niteroi, Brazil

RECOMB-CG 2012 (Poster)
Hernandez-Rosales M, Nicolas Wieseke, Marc Hellmuth, Peter F. Stadler.: *Simulation of Gene Family Histories.*
10/2012; Niteroi, Brazil

JOBIM 2011 (Poster)
Hernandez-Rosales M, Nicolas Wieseke, Marc Hellmuth, Peter F. Stadler.: *Simulation of Gene Family Histories.*
06/2011; Paris, France

Bioinformatics Autumn Seminar 2011 (Presenter)
Hernandez-Rosales M, Marc Hellmuth, Nicolas Wieseke, Katharina Huber, Vincent Moulton, and Peter F. Stadler.: *From Orthology Relations, through Symbolic Ultrametrics to Cographs.*
10/2011; Vysoka Lipa, Czech Republic.

26th TBI Winter Seminar 2011 (Presenter)
Hernandez-Rosales M: *Generation of Gene Family Histories.*
06/2011; Bled, Slovenia

Bioinformatics Autumn Seminar 2010 (Presenter)
Hernandez-Rosales M: *Reconstructing of Gene Family Histories.*
10/2010; Vysoka Lipa, Czech Republic

Bioinformatics Autumn Seminar 2009 (Presenter)
Hernandez-Rosales M: *On the reconstruction of the evolutionary history of gene families.*
10/2009; Vysoka Lipa, Czech Republic

23th TBI winter seminar 2008 (Presenter)
Hernandez-Rosales M: *Phylogenetics + Phyloenomics.*
02/2008; Bled, Slovenia

Bioinformatics Autumn Seminar 2007 (Presenter)
Hernandez-Rosales M: *Predicting ncRNA targets in Pseudonomas Aeruginosa.*
10/2007; Studeny, Czech Republic

GCB 2007 (Poster)
Sven Findeiss, Hernandez-Rosales M, Peter F. Stadler.: *BioInformatic approaches to detect and verify ncRNAs.*
09/2007; Bochum, Germany

RECOMB 2005 (Poster)
Hernandez-Rosales M, Sarath Janga.: *No evidence for relation between Co-regulation and Regulatory Interactions in the evolution of Regulatory Networks.*
05/2005; Boston, USA

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder Dienstleistungen als solche gekennzeichnet.

Leipzig, den 07. Juni 2013

(Maribel Hernandez Rosales)