Gene Family Histories

Theory and Algorithms

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

M. Sc. David Schaller

geboren am 01.06.1994 in Glauchau

Die Annahme der Dissertation wurde empfohlen von:

1. Prof. Dr. Peter F. Stadler, Universität Leipzig
   Prof. Dr. Marc Hellmuth, Stockholms universitet
2. Prof. Dr. Mike Steel, University of Canterbury

Die Verleihung des akademischen Grades erfolgt mit Bestehen
der Verteidigung am 22.10.2021 mit dem Gesamtprädikat *summa cum laude*.

David Schaller: *Gene Family Histories – Theory and Algorithms,* © *October 2021*

To my grandpas
Für meine Opas

ACKNOWLEDGMENTS

First of all, I want to thank my two supervisors Peter Stadler and Marc Hellmuth for the privilege of contributing to this project, for their excellent guidance, their trust in me, and of course their great interest in my work. I enjoyed working with you a lot and going through the proofs in endless video calls kept me motivated in these weird times of working from home.

Many thanks to Manuela Geiß for passing on her project to me, and also for staying a part of it, as well as to my other co-authors Nicolas Wieseke, Manuel Lafond, and the group around Maribel Hernández-Rosales, especially Ali López Sánchez and Marcos Lafitte for a wonderful time during their stay here.

I would like to thank my colleagues at the Bioinf, in particular, Petra Pregel for guiding me through the bureaucracy, and Carsten Seemann for many fruitful discussions and his contagious passion for math. I am looking forward to having cake and coffee breaks with all of you again as soon as the incidence value allows it (probably we all have quite some cake debt by now).

Furthermore, I want to thank the people at the Max Planck Institute for Mathematics in the Sciences in Leipzig, especially Antje Vandenberg, Jürgen Jost, and Bernd Sturmfels, and gratefully acknowledge the funding and scientific freedom granted to me by the Max Planck Society.

Thanks to my friends Sofia Haller and Robert Böttner for proofreading some chapters of this work (and for keeping the fundamental criticism to a minimum ;-)).

Finally, a huge thank you goes to my family, in particular, to my parents for supporting me wherever they could in all these years, and of course to my partner Kevin for being so patient and also for being impatient whenever he felt I needed some recreation or struggled with my usual indecisiveness. Thank you for all your love!

ABSTRACT

Detailed gene family histories and reconciliations with species trees are a prerequisite for studying associations between genetic and phenotypic innovations. Even though the true evolutionary scenarios are usually unknown, they impose certain constraints on the mathematical structure of data obtained from simple yes/no questions in pairwise comparisons of gene sequences. Recent advances in this field have led to the development of methods for reconstructing (aspects of) the scenarios on the basis of such relation data, which can most naturally be represented by graphs on the set of considered genes.

Graph-based detection of orthologs, i.e., related genes that diverged as a consequence of a speciation event, follow this paradigm since they usually start from estimations of (reciprocal) best matches being the evolutionarily closest relatives of a gene in another species. Orthology is a crucial concept in genome annotation and phylogenomics. However, the mathematical properties of best match graphs (BMGs) have only recently been investigated in detail. This work gives novel characterizations of this class of vertex-colored directed graphs and some interesting subclasses, and studies their explaining gene trees. In particular, BMGs that are associated with fully resolved trees are considered. All such binary trees are refinements of the unique binary-resolvable tree (BRT) which can be constructed in near-cubic time for a given binary-explainable BMG.

There are two main sources of error in pipelines for orthology inference based on BMGs. Firstly, measurement errors in the estimation of best matches from sequence similarity in general lead to violations of the characteristic properties of BMGs. This issue is addressed by means of arc modification problems. We show that BMG editing, completion, and deletion are all NP-complete in general, and provide integer linear programming formulations to obtain exact solutions at least for small instances. Furthermore, a class of heuristic approaches to BMG editing are studied that are similar to Aho et al.'s supertree algorithm, and operate in a top-down fashion on a special set of triples extracted from the input graph. Benchmarking results suggest that the underlying tree structure is preserved in (moderately) perturbed BMGs. The second issue concerns the reconstruction of the orthology relation from a valid BMG. It was shown previously that, in the absence of horizontal gene transfer (HGT), every pair of orthologs forms a reciprocal best match. We extend these results by characterizing unambiguously false-positive (*u-fp*) orthology assignments in the setting of duplication-loss scenarios. In particular, these edges can be identified in polynomial time.

Several implicit methods for HGT inference focus on pairs of genes that have diverged only after the divergence of the two species in which the genes reside. This situation defines the edge set of an undirected graph, the later-divergence-time (LDT) graph. We investigate these graphs in the setting of relaxed scenarios, i.e., evolutionary scenarios that encompass all commonly used variants of duplication-transfer-loss scenarios in the literature. We characterize LDT graphs as a subclass of properly vertex-colored cographs that can be recognized in polynomial time, and provide an algorithm to construct a relaxed scenario that explains a given LDT graph. An edge in an LDT graph implies that the two corresponding genes are separated by at least one HGT event. The converse is not true, however. We introduce rs-Fitch graphs which formalize the complete xenology relation. This class of vertex-colored graphs comprises complete multipartite graphs that satisfy certain constraints on the vertex colorings and is also recognizable in polynomial time. Finally, the question of how much information about all HGT events is contained in LDT graphs is addressed with the help of simulations of evolutionary scenarios with a wide range of duplication, loss, and HGT event rates.

CONTENTS

1 INTRODUCTION 3

2 BASIC DEFINITIONS 7

2.1 Sets and Binary Relations . . . 7
2.2 Graphs . . . 8
2.2.1 Adjacency, Subgraphs, and Connectedness . . . 8
2.2.2 Special Graphs, Graph Properties, and Operations . . . 9
2.2.3 Colorings . . . 10
2.3 Trees . . . 10
2.3.1 Tree Isomorphisms and Clusters . . . 11
2.3.2 Last Common Ancestors . . . 12
2.3.3 Restrictions and Refinements . . . 12
2.3.4 Rooted Triples and Triple Consistency . . . 13
2.3.5 The Span and the Closure Operator . . . 16
2.3.6 Special Trees . . . 16
2.4 Cographs & Cotrees . . . 17
2.5 Computational Complexity . . . 18

3 CONCEPTS AND METHODS IN PHYLOGENETICS 21

3.1 Evolutionary Scenarios and the Types of Homology . . . 21
3.2 The Biology Behind Evolutionary Events . . . 25
3.3 Inference Methods . . . 28
3.3.1 Gene Similarity, Distances, and the Molecular Clock . . . 28
3.3.2 Inference of Trees . . . 30
3.3.3 Inference of Orthology . . . 34
3.3.4 Inference of Horizontal Gene Transfer . . . 38
3.4 Simulation of Phylogenetic Scenarios . . . 39

4 BEST MATCH GRAPHS 43

4.1 Recent Work on Best Match Graphs . . . 44
4.1.1 Best Matches, BMGs, and Their Basic Properties . . . 44
4.1.2 Least Resolved Trees . . . 47
4.1.3 Reciprocal Best Match Graphs . . . 47
4.1.4 Inference of Best Matches . . . 48
4.2 Novel Characterizations of Best Match Graphs . . . 50
4.2.1 Useful Properties . . . 50
4.2.2 Least Resolved Trees . . . 50
4.2.3 Characterization via Informative Triples . . . 55
4.2.4 Characterization via Informative and Forbidden Triples 59
4.3 Binary-Explainable Best Match Graphs . . . 62
4.3.1 Hourglasses . . . 63
4.3.2 Binary Trees Explaining a BMG in Near Cubic Time . . . 67

4.3.3 The Binary-Resolvable Tree 70
4.3.4 Simulation Results 73
4.4 Two-Colored Best Match Graphs 76
4.4.1 Recent Work on 2-BMGs 76
4.4.2 Forbidden Induced Subgraphs of 2-BMGs 78
4.4.3 Support Leaves 81
4.4.4 Construction of LRTs of 2-BMGs 85
4.4.5 Binary-Explainable 2-BMGs 92
4.5 Summary 93

5 BEST MATCH GRAPH EDITING 95

5.1 BMG Modification Problems 95
5.2 Complexity of 2-BMG Modification Problems 97
5.3 Complexity of ℓ-BMG Modification Problems 110
5.4 ILP Formulation of BMG Modification Problems 113
5.4.1 2-BMG Modification Problems 114
5.4.2 General ℓ-BMG Modification Problems 115
5.4.3 Binary-Explainable BMG Modification Problems 116
5.4.4 Remarks on the Running Time 117
5.5 Heuristics of Best Match Graph Editing 118
5.5.1 A Simple, Triple-Based Heuristic 119
5.5.2 Locally Optimal Splits 121
5.5.3 Minimizing the *UR*-Cost $c(\vec{G},\mathcal{V})$ 131
5.5.4 Computational Experiments 134
5.6 Summary 149

6 FROM BEST MATCHES TO ORTHOLOGY 151

6.1 Reconciliation Maps, Event-Labeling, and Orthology Relations 152
6.2 False-Positive Orthology Assignments 156
6.3 Characterization of Unambiguous False-Positive (*u-fp*) Edges . 159
6.3.1 *u-fp* Edges in Quartets 161
6.3.2 *u-fp* Edges in Hourglasses and Hourglass Chains 165
6.3.3 Color-set Intersection Graphs 169
6.3.4 Hug-Edges and No-Hug Graphs 171
6.3.5 Resolving Least Resolved Trees 173
6.3.6 Extremal Labeling of Augmented Trees 177
6.3.7 Additional Unidentified False Positives 179
6.4 Quartets, Hourglasses, and the Structure of Reciprocal Best Match Graphs 183
6.4.1 Hourglass-Free BMGs 183
6.4.2 *u-fp* Edges in Hourglass Chains 185
6.4.3 Four-Colored P_4s 185
6.5 Simulation Results 188
6.6 Summary 191

7 LATER-DIVERGENCE-TIME (LDT) GRAPHS 193

7.1 Relaxed Reconciliation Maps 194

7.2 LDT Graphs and μ-free Scenarios 196
7.3 Properties of LDT Graphs 201
7.4 Recognition and Characterization of LDT Graphs 204
7.5 Least Resolved Trees for LDT graphs 214
7.6 LDT Graph Editing 215
7.7 Summary 217

8 FROM LDT GRAPHS TO XENOLOGY 219

8.1 HGT-Labeled Trees and rs-Fitch Graphs 219
8.2 LDT Graphs and rs-Fitch Graphs 222
8.3 rs-Fitch Graphs with General Colorings 226
8.4 Least Resolved Trees for Fitch graphs 232
8.5 Editing Problems 237
  8.5.1 Editing Colored Graphs to rs-Fitch Graphs 237
  8.5.2 Editing LDT Graphs to Fitch Graphs 237
8.6 Simulation Results 242
8.7 Summary 246

9 CONCLUSION 247

PUBLICATIONS

Some results, ideas, and figures in this work have appeared previously in the following publications:

1. D. Schaller, M. Geiß, P.F. Stadler, M. Hellmuth (2021) "Complete Characterization of Incorrect Orthology Assignments in Best Match Graphs." In: *Journal of Mathematical Biology*, 82(3):20, doi: 10.1007/s00285-021-01564-8
2. D. Schaller, P.F. Stadler, M. Hellmuth (2021) "Complexity of modification problems for best match graphs." In: *Theoretical Computer Science*, 865:63-84, doi: 10.1016/j.tcs.2021.02.037
3. D. Schaller, M. Geiß, E. Chávez, M.G. Lafitte, A. López Sánchez, B.M. Stadler, D.I. Valdivia, M. Hellmuth, M. Hernández-Rosales, P.F. Stadler (2021) "Corrigendum to "Best match graphs"." In: *Journal of Mathematical Biology*, 82(6):47, doi: 10.1007/s00285-021-01601-6
4. D. Schaller, M. Geiß, M. Hellmuth, P.F. Stadler (2021) "Best Match Graphs with Binary Trees." In: C. Martín-Vide, M.A. Vega-Rodríguez, T. Wheeler (Eds.), *Algorithms for Computational Biology, 8th AlCoB*, volume 12715 of *Lecture Notes in Computer Science*, pages 82-93, doi: 10.1007/978-3-030-74432-8_6
5. D. Schaller, M. Lafond, P.F. Stadler, N. Wieseke, M. Hellmuth (2021) "Indirect identification of horizontal gene transfer." In: *Journal of Mathematical Biology*, 83(1):10, doi: 10.1007/s00285-021-01631-0
6. D. Schaller, M. Geiß, M. Hellmuth, P.F. Stadler (2021) "Least resolved trees for two-colored best match graphs." In: *Journal of Graph Algorithms and Applications*, 25(1):397-416, doi: 10.7155/jgaa.00564
7. D. Schaller, M. Geiß, M. Hellmuth, P.F. Stadler (2021) "Heuristic algorithms for best match graph editing." In: *Algorithms for Molecular Biology*, 16(1):19, doi: 10.1186/s13015-021-00196-3
8. D. Schaller, M. Geiß, M. Hellmuth, P.F. Stadler (2021) "Arc-Completion of 2-Colored Best Match Graphs to Binary-Explainable Best Match Graphs." In: *Algorithms*, 14(4):110, doi: 10.3390/a14040110

A complete list of all my publications can be found at the end of this thesis.

1

INTRODUCTION

"I think"

Two words written above the first known illustration of an evolutionary tree. The famous sketch was drawn by Charles Darwin in his "Notebook B" on the *Transmutation of species* in 1837 [322]. By then, he had understood that the diversity of species resulted from gradual processes of change, selection, and branching from a common ancestor – a theory known today as *evolution*. His book *On the Origin of Species* was published in 1859, and before long, the majority of scientists were convinced that evolution was real, despite many controversies surrounding its exact mechanisms [33].

Phyolgenetics is the field of biology that addresses the reconstruction of histories and relationships of biological entities such as species. In Darwin's time, the DNA as the main molecule carrying the hereditary information for the development of organisms had not yet been discovered. As a consequence, taxonomists mainly relied on the comparison of morphological and physiological traits and, in some cases, fossils to infer the relationships of species. The development of high-throughput methods for DNA sequencing and, as a result, the ever-increasing availability of sequenced genomes in the last few decades have led to the dominance of *molecular phylogenetics*, a branch that reconstructs species trees or networks based on sequence comparisons and sophisticated mathematical models for their evolution [88, 335]. Intriguingly, these advances enable us to observe ongoing evolution within the life span of a human, especially on the border of living and dead matter – in the kingdom of viruses. This is more relevant than ever since the appearance of new mutated lineages of SARS-CoV-2 poses a threat in the current pandemic situation e.g. to the effectiveness of vaccines [94, 314, 321].

Around the same time as Darwin, the Augustinian friar Gregor Mendel proposed the existence of discrete heritable traits. However, his groundbreaking findings remained largely unappreciated for decades and he was even accused of falsifying his data [327]. Today, these traits are known as genes and are usually defined as genomic sequences that encode some kind of functional molecule [105]. Since the members of a so-called gene family share a common ancestry and therefore usually show a certain degree of sequence similarity, they are of key importance for all sequence-based analyses.

The evolution of a gene family is closely tied to that of the species in which they reside. In particular, whenever a species splits into two or more lineages that become reproductively isolated, the genes also continue to evolve independently. Two genes in different species that have diverged as a consequence of such a *speciation* event have been termed *orthologs* [89]. The knowledge of orthologous genes is of particular practical interest in computational biology for two main reasons: (i) the time since their divergence coincides with that of their corresponding species, and thus, they are essential for the reconstruction of species trees based on sequence similarity, and (ii) they are usually assumed to fulfill the same or similar functions across species [309]. Albeit controversial, the latter hypothesis is widely made in the computational prediction of gene functions [95, 222, 296]. Not all genes that share a common ancestry are orthologs. This is due to other types of events that shape the history of a gene family. Genes that arose from a *duplication* event, i.e., the insertion of a copy of a genomic region into the same genome, are called *paralogs* [89]. Similarly, genetic material can be transferred from one species to another – a process called *horizontal gene transfer (HGT)*. This process is most common in prokaryotes, where it e.g. plays a crucial role in the acquisition of antibiotic resistance genes [209], but also contributed to the genomes of extant eukaryotes [63, 291]. Two genes of which at least one has undergone a horizontal transfer since their divergence are called *xenologs* [91]. The fourth of the most common events in gene families (and of those that will be considered here) is *gene loss*, i.e., a gene vanishes from the genome as a consequence of a single mutational event or the gradual accumulation of smaller mutations [8].

As a consequence of the different types of events, gene trees, representing the evolutionary history of a gene family, are often incongruent with the corresponding species trees. They are, however, "embedded" into the species trees in a way that branching events corresponding to speciations are tied to the branchings in the species trees, whereas all other types of events are located independently of the latter. Moreover, genes can only be transferred between species that coexisted at a certain point in their evolution which imposes certain time-consistency constraints on the evolutionary history [108, 212, 228]. A major task in phylogenetics, apart from the reconstruction of tress, is therefore the constructions of so-called *reconciliations* which are formalizations of the notion of an embedding of one tree into another, see [80, 106, 109, 238, 316] for some examples.

Detailed gene family histories and reconciliations are a valuable tool for studying associations between genetic and phenotypic innovations. In particular, the orthology, paralogy, and xenology relation on the set of genes under consideration can be extracted as a by-product from a reconciliation. Tree-based methods for orthology and HGT inference indeed follow this strategy, see [15] and [246] for reviews. However, they suffer from difficulties in the reconstruction of species and gene trees which usually have to assume that sequence evolution approximately follows some well-behaved mathematical model [88, 335], as well as high computational costs of finding reconciliations based on some optimality criterion [178].

Considerable efforts have therefore been put into the development of methods that circumvent the necessity of explicitly constructing trees and reconciliations. Graph-based methods for orthology inference start from *best matches*, i.e., by identifying, for each gene, its closest relative or relatives in every other species, see e.g. [226, 252, 272] for a few key reviews and the references therein. Similarly, implicit phylogenetic methods for HGT detection identify xenologs as such genes whose divergence time differs from that of their respective species. This is approximated by significantly lower or higher sequence similarity as compared to the majority of pairs of related genes [246, 254].

The tree-less inference of such relation data also motivated a series of recent publications that are concerned with the reversed procedure, i.e., how evolutionary histories can be reconstructed from estimates of the orthology, paralogy, or xenology relation. In particular, the mathematical structure of orthology [132, 180], xenology [101, 131, 135], and the best match relation [102] have been characterized which led to efficient algorithms to extract gene trees, species trees, and reconciliations from the relation data. Although the resulting representations of gene family histories are usually not fully resolved, they can provide important constraints for subsequent refinements. The advantage of the relation-based approach is primarily robustness [133, 293] since it starts from yes/no answers to simple, pairwise comparisons instead of detailed probabilistic models. In particular, the data is a binary relation, and thus, it can be represented by a graph, possibly augmented by a measure of confidence. Noise and inaccuracies in the initial estimates then translate into violations of the required mathematical properties of the graphs in question. Therefore, graph editing approaches can be harnessed as a means of noise reduction [79, 133, 137, 185].

This work is a contribution to a detailed mathematical understanding of the different relations between genes and their interdependencies. It focuses on two graphs: *best match graphs (BMGs)* and *later-divergence-time (LDT) graphs*, which provide a theoretical framework for graph-based inference of the orthology and xenology relation, respectively.

Best match graphs (BMGs) have only very recently been introduced as mathematical objects to formalize the idea of pairs of evolutionarily most closely related genes in two species [102]. In practice, sequence similarity is usually used as a proxy for relatedness. Stadler et al. [293] investigated in some detail how best matches in the somewhat stricter sense of ancestry in the gene tree can be reliably obtained from sequence data. As alluded to above, best matches are closely related to orthology. In particular, two orthologous genes are always best matches of one another (so-called reciprocal best matches), at least in the absence of horizontal gene transfer [103]. In other words, there are no false-negative orthology assignments when reciprocal best matches are considered. Here, we investigate this relationship in some more depth and ask which reciprocal best matches can be identified as false-positive orthologs with the knowledge of the full BMG, i.e., by also considering the unidirectional best matches. Moreover, we continue the study of how much information on the gene tree is contained in a BMG, and ask what can be inferred if we make additional assumptions such as an evolution that only produces bifurcations in the gene tree.

Horizontal gene transfer largely invalidates mathematical guarantees for the relationship of orthology and best matches [103]. It is therefore of interest to first identify groups of genes that do not contain transfers in their histories. While an unexpectedly large divergence time of two genes as inferred in implicit phylogenetic methods for HGT detection can alternatively be explained by duplication and loss events, a smaller divergence time is a clear indicator of xenologs. To the best of our knowledge, there has not been an in-depth investigation of the mathematical properties of such "later-divergence-time" data. Here, we introduce LDT graphs as a formalization of this notion and study the scenarios that explain them. In particular, we answer the question of how LDT graphs can be recognized and when they coincide with the xenology relation. Since the latter is not satisfied in general, we also propose methods for inferring the remaining xenologs by means of completing an LDT graph to a so-called Fitch graph [135], and explore their performance with the help of simulations.

STRUCTURE OF THIS THESIS

Chapter 2 introduces the basic mathematical definitions, the notation and some well-established results that will be needed throughout this work. This is followed by an overview of the concepts and state-of-the-art methods in the field of phylogenetics in Chapter 3. In particular, orthology, paralogy, and xenology are formally introduced as the different types of homology.

Best match graphs and their properties are extensively revisited in Chapter 4. Several novel characterizations are given for them as well as some interesting subclasses. In particular, BMGs that are associated with fully resolved trees are studied. The question of how to correct initial estimates of best match data is tackled in Chapter 5 using graph modification problems. Chapter 6 then turns to the question of how the orthology relation can be extracted from a BMG. To this end, only duplication-loss scenarios are considered, i.e., horizontal gene transfer is explicitly excluded. In this setting, one obtains a characterization of all reciprocal best matches that can unambiguously be identified as false positives w.r.t. orthology with the knowledge of a BMG alone. Chapter 7 introduces later-divergence-time graphs in the framework of relaxed scenarios, a very general version of reconciliation that primarily focuses on time consistency. Finally, the relationship between LDT graphs and xenology is studied in Chapter 8.

2

BASIC DEFINITIONS

IN THIS CHAPTER, the notation and basic mathematical concepts are introduced that will be used throughout this work. Secs. 2.1–2.3 contain rather general definitions concerning sets, binary relations, graphs, and trees. Mostly, we will adopt the standard notation established in graph theory (see e.g. [78]) and phylogenetics (see e.g. [271]). In Sec. 2.4, we turn to the special class of cographs and their representations as cotrees. We close in Sec. 2.5 with a brief introduction into computational complexity. We note that we will refer to some concepts from this latter section earlier in the chapter.

2.1 SETS AND BINARY RELATIONS

Throughout, we will only consider finite sets (with obvious exceptions such as the real numbers $\mathbb{R}$). Given two sets A and B, we write $A \subseteq B$ ($A \subset B$) if A is a (proper) subset of B. In case of a proper subset, we sometimes also write $A \subsetneq B$ to emphasize that $A \neq B$. The union, the intersection, and the set difference will be denoted by $A \cup B$, $A \cap B$, and $A \setminus B$, respectively. Moreover, we write $A_1 \mathbin{\dot\cup} \dots \mathbin{\dot\cup} A_k$ for the (disjoint) union to emphasize that the sets $A_1, \dots, A_k$, $k \geq 2$, are pairwise disjoint. We denote by $A \mathbin{\triangle} B := (A \setminus B) \mathbin{\dot\cup} (B \setminus A)$ the symmetric difference of two sets A and B. The *power set* of A is the set of all subsets of A (including $\emptyset$ and A), and denoted by 2^A. The set of (unordered) n-element subsets of A is given by $\binom{A}{n} := \{A' \subseteq A \colon |A'| = n\} \subseteq 2^A$. The *Cartesian product* of two sets A and B is $A \times B := \{(a,b) \mid a \in A \text{ and } b \in B\}$. We will, moreover, frequently need the (irreflexive part of the) set of ordered pairs of the set A, i.e., $(A \times A)_{\mathrm{irr}} := A \times A \setminus \{(a,a) \mid a \in A\}$.

A *partition* $\mathcal{A} = \{A_1, \dots, A_k\}$ with $k \geq 1$ of a non-empty set A satisfies (i) $A_i \neq \emptyset$ for $1 \leq i \leq k$, (ii) $\bigcup_{i=1}^{k} A_i = A$, and (iii) $A_i \cap A_j = \emptyset$ for $1 \leq i < j \leq k$. A partition is non-trivial if $k = |\mathcal{A}| \geq 2$. Consider two partitions $\mathcal{A} = \{A_1, \dots, A_k\}$ and $\mathcal{A}' = \{A'_1, \dots, A'_l\}$ of A. If for every $1 \leq j' \leq l$ there is a j such that $A'_{j'} \subseteq A_j$, i.e., if every set in $\mathcal{A}'$ is completely contained in a set in $\mathcal{A}$, then $\mathcal{A}'$ is a *refinement* of $\mathcal{A}$, and $\mathcal{A}$ is a *coarse-graining* of $\mathcal{A}'$.

A collection $\mathcal{C} \subseteq 2^A$ of non-empty subsets of A is a *hierarchy* if (i) $\{a\} \in \mathcal{C}$ for all $a \in A$, (ii) $A \in \mathcal{C}$, and (iii) $A_i \cap A_j \in \{\emptyset, A_i, A_j\}$ for all $A_i, A_j \in \mathcal{C}$. In other words, Condition (iii) states that all pairs of sets in $\mathcal{C}$ must either be disjoint or in a subset relation.

A *(binary) relation* on a set A is a set $R \subseteq A \times A$ of ordered pairs (a,b) with $a,b \in A$. Instead of $(a,b) \in R$, we will also write aRb. The relation R is *symmetric* if $(a,b) \in R$ implies $(b,a) \in R$ for all $a,b \in A$, in which case we also write $ab \in R$. It is *antisymmetric* if $(a,b),(b,a) \in R$ implies $a = b$. It is *reflexive* (*irreflexive*) if $(a,a) \in R$ ($(a,a) \notin R$) holds for all $a \in A$. Furthermore, R is *transitive* if $(a,b),(b,c) \in R$ implies $(a,c) \in R$ for all (not necessarily distinct) $a,b,c \in R$. A relation R that is symmetric, reflexive and transitive is called an *equivalence relation*. In this case, the *equivalence class* of $a \in A$ is $[a] := \{b \in A \mid ab \in R\}$, and the set of all equivalence classes forms a partition of A. A relation that is reflexive, antisymmetric and transitive is a *partial order.*

2.2 GRAPHS

A *graph* $G = (V, E)$ is an ordered pair of the two sets V, the *vertices* (or *nodes*), and E, the *edges* or *arcs* depending on the type of the graph. We will also write $V(G)$ and $E(G)$, resp., to refer to a specific graph G. We will consider *undirected* and *directed* graphs. An undirected graph satisfies $E \subseteq \binom{V}{2}$. We will call the elements $\{x, y\} \in E$ *edges* and simply write xy (or equivalently yx). A directed graph (or *digraph*), on the other hand, has a set of (directed) *arcs* $E \subseteq (V \times V)_{\mathrm{irr}}$, and will often be denoted by $\vec{G}$. An arc $(x, y) \in E$ is also called an arc from x to y, an *out-arc* of x and an *in-arc* of y. Note that both types of graphs are defined in a way that excludes (self) loops, i.e., edges or arcs connecting a vertex with itself. In the literature, such graphs are usually called *simple*.

A graph $G' = (V', E')$ is a *subgraph* of $G = (V, E)$, in symbols $G' \subseteq G$, if $V' \subseteq V$ and $E' \subseteq E$. The subgraph is called *induced* if, additionally, $xy \in E$ (or, in case of a digraph, $(x, y) \in E$) implies $xy \in E'$ ($(x, y) \in E'$) for all $x, y \in V'$. The subgraph of G that is induced by a subset of vertices $W \subseteq V$ is denoted by $G[W]$.

The *underlying undirected graph* G of a digraph $\vec{G}$ can be obtained from $\vec{G}$ by "dropping the direction" of all arcs in $E(\vec{G})$, i.e., setting $xy \in E(G)$ if an only if $(x, y) \in E(\vec{G})$ or $(y, x) \in E(\vec{G})$ (or both). The (undirected) *symmetric part* G of a digraph $\vec{G}$ also has vertex set $V(\vec{G})$ but edges $xy \in E(G)$ if and only if $(x, y), (y, x) \in E(\vec{G})$, i.e., (x, y) and (y, x) are *bidirectional* arcs in $\vec{G}$. The class of undirected graphs G can alternatively by defined as the subclass of directed graphs that satisfy $(x, y) \in E(G)$ if and only if $(y, x) \in E(G)$ for all $x, y \in V(G)$. With this definition, the symmetric part is always a subgraph of the original graph. We will make use of this definition in Chapter 6.

Both reflexive and irreflexive binary relations R (on V) can be represented by a directed graph $\vec{G} = (V, E := R \setminus \{(v, v \mid v \in V)\})$. If R is additionally symmetric, it can analogously be represented by an undirected graph.

2.2.1 *Adjacency, Subgraphs, and Connectedness*

Two vertices in a (di)graph G are *adjacent* if they are connected by an edge (an arc). A vertex x and an edge yz (or an arc (y, z)) are *incident* if $x \in \{y, z\}$. The *neighbors* or *neighborhood* of x in an undirected graph, denoted by $N(x)$, is the set of all vertices that are adjacent to x, its *degree* is $\mathsf{deg}(x) := |N(x)|$. In case of a directed graph, we distinguish *out-neighbors* (or the *out-neighborhood*) $N^+(x) := \{y \in V \mid (x, y) \in E\}$ and *in-neighbors* (or the *in-neighborhood*) $N^-(x) := \{y \in V \mid (y, x) \in E\}$ of a vertex x, as well as its *out-degree*, $\mathsf{outdeg}(x) := |N^+(x)|$, and *in-degree*, $\mathsf{indeg}(x) := |N^-(x)|$. To keep the notation simple, we put here $N(x) := N^+(x)$. Consequently, for subsets

$W \subseteq V$, we also write $N(W) := \bigcup_{x \in W} N(x) \setminus W$ for both types of graphs, and additionally $N^{-}(W) := \bigcup_{x \in W} N^{-}(x) \setminus W$ in case of a digraph.

A sequence $S = (x_1, \ldots, x_n)$ of vertices in an undirected graph $G = (V, E)$ is a *path* (of length $n-1$) if (i) all elements in S are pairwise distinct, and (ii) $x_i x_{i+1} \in E$ holds for all $1 \leq i \leq n-1$. Moreover, S is a circle if additionally $x_1 x_n \in E$. The sequence S is a *shortest part* if there is no other path in G connecting x_1 and x_n that contains less vertices. In a digraph $\vec{G} = (V, E)$, $S = (x_1, \ldots, x_n)$ is a *(directed) path* if (i) all elements in S are pairwise distinct, and (ii) $(x_i, x_{i+1}) \in E$ holds for all $1 \leq i \leq n-1$, and a *(directed) circle* if additionally $(x_n, x_1) \in E$.

Two vertices are *connected* in an undirected graph $G = (V, E)$ if there exists a path that connects them. If this is the case for any two vertices in G, then G is called *connected*, and *disconnected* otherwise. A *connected component* C is a connected subgraph of G that is maximal w.r.t. inclusion, i.e., there is no vertex $x \in V \setminus V(C)$ such that $G[V(C) \cup \{x\}]$ is connected. Whenever the context is clear, we will also refer to the vertex sets of these inclusion-maximal subgraphs as the set $\mathcal{C}$ of connected components. In this case, $\mathcal{C}$ is a partition of the vertex set V. Furthermore, for connected undirected graphs G, we define the *diameter*, denoted by $\operatorname{diam}(G)$, as the length of the longest shortest path between any two vertices of G.

In a digraph $\vec{G}$, two vertices x and y are *strongly connected* if they are connected by a directed path of the form $(x, \ldots, y)$, and *weakly connected* if they are connected in the underlying undirected graph of $\vec{G}$. Correspondingly, $\vec{G}$ is *strongly* or *weakly connected*, resp., if this is the case for any two of its vertices, and we can define *strongly* or *weakly connected components* as inclusion-maximal strongly or weakly connected subgraphs of $\vec{G}$, respectively. We usually call a weakly connected digraph/component simply connected. Hence, when referring to a digraph, *disconnected* means that it is not weakly connected.

2.2.2 *Special Graphs, Graph Properties, and Operations*

An undirected graph $G = (V, E)$ with $E = \binom{V}{2}$ is called *complete* and denoted by $K_{|V|}$. A complete subgraph $G' \subseteq G$ is called a *clique* in G. A graph whose connected components are all cliques is a *cluster graph*. A subset of vertices W is an *independent set* in a directed or undirected graph G if $G[W]$ is edgeless and W is maximal w.r.t. inclusion. It will be useful here to include the maximality condition even though it usually is not part of the definition of independent sets.

We will denote by P_n and C_n subgraphs with n vertices in an undirected graph that form a path or circle, resp., that is, with no additional edges.

In a digraph, a vertex x is a *sink* if $N(x) = \emptyset$, and a *source* if $N^{-}(x) = \emptyset$. A digraph $\vec{G}$ is *sink-free* if it has no sinks. We call x a *hub-vertex* of a digraph $\vec{G} = (V, E)$ if $(x, y), (y, x) \in E$ holds for all vertices $y \in V \setminus \{x\}$.

A directed or undirected graph $G = (V, E)$ is called *k-partite* if there exists a partition $\mathcal{V} = \{V_1, \ldots, V_k\}$ with $k \geq 1$ such that $x, y \in V_i \in \mathcal{V}$ implies $xy \notin E$ (or $(x, y), (y, x) \notin E$ in case of a digraph). For the special case that $k = 2$, we also use the term *bipartite*, and *multipartite* for an arbitrary k. If G is k-partite with independent sets $\mathcal{V} = \{V_1, \dots, V_k\}$ and, additionally, $x \in V_i, y \in V_j$ with $i \neq j$ implies $xy \in E$ (or $(x,y),(y,x) \in E$), then G is called *complete multipartite*. Note that, since we allow $k = 1$, every edgeless graph is always complete multipartite.

The *complement* of an undirected graph $G = (V,E)$, denoted by $\overline{G}$, has vertex set V and edge set $\overline{E} := \binom{V}{2} \setminus E$. Given two undirected graphs $G = (V,E)$ and $H = (W,F)$ that satisfy $V \cap W = \emptyset$, their *disjoint union* is given by $G \mathbin{\dot\cup} H := (V \mathbin{\dot\cup} W, E \mathbin{\dot\cup} F)$, and their *join* by $G \triangledown H := (V \mathbin{\dot\cup} W, E \mathbin{\dot\cup} F \mathbin{\dot\cup} \{vw \mid v \in V, w \in W\})$. In Chapters 7 and 8, we will need the graph $K_2 + K_1$ which is the disjoint union of a K_2 and a K_1.

We write $G - W := G[V \setminus W]$ for the graph that is obtained from G by removal of all vertices in W and all edges/arcs incident to elements in W. Similarly, for a graph $G = (V,E)$ and edge set $F \subseteq \binom{V}{2}$ (or, in case of a digraph, arc set $F \subseteq (V \times V)_{\text{irr}}$), we define the graphs $G + F := (V, E \cup F)$, $G - F := (V, E \setminus F)$, and $G \triangle F := (V, E \triangle F)$.

2.2.3 *Colorings*

Given a graph $G = (V,E)$, a map $\sigma \colon V \to M$ where M is a non-empty set of *colors* is called a *(vertex-)coloring*. A graph G endowed with a coloring σ, or short (vertex-)colored graph, will be denoted by (G,σ). The coloring σ is *proper* if $\sigma(x) \neq \sigma(y)$ for any two adjacent vertices $x, y \in V$. As a consequence, we have

Observation 2.1. *Let x be a hub-vertex in a properly colored digraph $(\vec{G},\sigma)$. Then x is the only vertex of color $\sigma(x)$ in $(\vec{G},\sigma)$.*

We write $\sigma(W) := \{\sigma(x) \mid x \in W\}$ for the subset of colors represented in $W \subseteq V$. Moreover, we denote by $V[r] := \{x \in V \mid \sigma(x) = r \text{ and } r \in M\}$ the subset of vertices of color r. We often write $|\sigma(V)|$-coloring to emphasize the number of colors in (G,σ). Throughout, we will need restrictions of the color map σ. For later reference, we present them in

Definition 2.1. *Let $\sigma \colon V \to M$ be a map and $V' \subseteq V$. The map $\sigma_{|V'} \colon V' \to M$ is defined by putting $\sigma_{|V'}(x) = \sigma(x)$ for all $x \in V'$.*

Given a set M' with $\sigma(V') \subseteq M' \subseteq M$, we may additionally restrict the codomain by writing $\sigma_{|V',M'} \colon V' \to M'$.

We will usually only restrict the domain of σ. Hence, the subgraph of a vertex-colored graph $(G = (V,E),\sigma)$ induced by $V' \subseteq V$ is given by $(G[V'],\sigma_{|V'})$. If not stated otherwise, we do neither assume that σ nor that its restrictions $\sigma_{|V'}$ or $\sigma_{|V',M'}$ are surjective.

2.3 TREES

An undirected graph without cycles is a *forest*. If it is additionally connected, it is called a *tree*. A *rooted tree* $T = (V,E)$ is a tree with one distinguished vertex $\rho_T \in V$ (or simply ρ if there is no risk of ambiguity) which is called the *root* of T. In this work, we will only consider rooted trees, and thus, simply write trees. Moreover, we assume that all trees are non-empty, i.e., V contains at least the root ρ_T.

The notion of a root naturally gives rises to the so-called *ancestor order*, a relation on V denoted by $\preceq_T$, that is given by $v \preceq_T u$ if and only if u lies along the unique path from v to the root ρ_T. One easily verifies that $\preceq_T$ is a partial order. If $v \preceq_T u$, we call u an *ancestor* of v, and v a descendant of u. If moreover $v \neq u$, we may write $v \prec_T u$ and speak of a *strict ancestor/descendant*. For convenience, we may also write $u \succeq_T v$ and $u \succ_T v$ instead of $v \preceq_T u$ and $v \prec_T u$, respectively. If $v \preceq_T u$ or $u \preceq_T v$, we say that u and v are *comparable* (in T), and *incomparable* otherwise. If $v \preceq_T u$ and $uv \in E$, then u is the (unique) *parent* of v, denoted by $\mathsf{par}_T(v)$, and v is a *child* of u. The set of all *children* of a vertex $u \in V$ will be denoted by $\mathsf{child}_T(u)$. For the edges $uv \in E$, we use the convention that $v \prec_T u$, i.e., v is a child of u.

The *leaf set* $L := L(T) \subseteq V$ are those vertices in a tree T with an empty set of children. For brevity, we will often write a *tree on* L denoting a tree with leaf set L. The vertices in $V^0(T) := V \setminus L$ are called the *inner vertices*. In particular, ρ_T is an inner vertex if and only if the tree has at least two vertices. An edge uv is an *inner edge* if both u and v are inner vertices, and an *outer edge* if one of them is a leaf.

A tree is *phylogenetic* if every inner vertex has at least two children, and *binary* if every inner vertex has exactly two children.

Remark 2.1. *If not explicitly stated otherwise, we will always assume that a tree is phylogenetic throughout this work.*

The (maximal) subtree of a tree T rooted at a vertex u is induced by the set of vertices $\{x \in V(T) \mid x \preceq_T u\}$, and will be denoted by $T(u)$. In particular, the set of all leaves that are descendants of a vertex u is given by $L(T(u))$.

We will often consider *leaf-colored trees* (T, σ), i.e., trees T endowed with a (color) map $\sigma\colon L(T) \to M$ where M is again a non-empty set of colors. The subtree of a leaf-colored tree (T, σ) rooted at $u \in V(T)$ is given by $(T(u), \sigma_{|L(T(u))})$.

2.3.1 *Tree Isomorphisms and Clusters*

We call two rooted trees T and T' on the same leaf set L *isomorphic* if there exists a bijection $\Phi\colon V(T) \to V(T')$ satisfying (i) $uv \in E(T)$ if and only if $\Phi(u)\Phi(v) \in E(T')$ for any $u, v \in V(T)$, (ii) $\Phi(\rho_T) = \rho_{T'}$, and (iii) $\Phi(x) = x$ for all $x \in L$. *For all purposes in this work, it will be convenient to consider two isomorphic trees as identical. In particular, uniqueness of a tree w.r.t. some property means that there exists no non-isomorphic tree satisfying this property.*

The set of *clusters* of a tree $T = (V, E)$ on L is given by $\mathcal{C}(T) := \{L(T(u)) \mid v \in V(T)\}$, and is always a hierarchy on L. Since rooted phylogenetic trees satisfy $L(T(u)) = L(T(v))$ if and only if $u = v$ for all $u, v \in V$, there exists a well-known one-to-one correspondence between rooted phylogenetic trees and hierarchies (on the same set L) [270].

2.3.2 *Last Common Ancestors*

Given a tree $T = (V, E)$, we extend the ancestor order to the union of vertices and edges by setting, for each edge $e = xy \in E$ with $y \prec_T x$ and vertex $v \in V$, $v \prec_T e$ if and only if $v \preceq_T y$ and $e \prec_T v$ if and only if $x \preceq_T v$. Similarly, for two edges $e = xy$ and $f = ab$ with $y \prec_T x$, $b \prec_T a$ in T, we put $e \preceq_T f$ if and only if $y \preceq_T b$.

For a non-empty subset $A \subseteq V \cup E$, we define $\operatorname{lca}_T(A)$, the *last common ancestor* of A, to be the unique $\preceq_T$-minimal vertex of T, i.e., the one that is furthest away from ρ_T, that is an ancestor of every $x \in A$. For simplicity, we simply write $\operatorname{lca}_T(x_1, \dots, x_n)$ instead of $\operatorname{lca}_T(\{x_1, \dots, x_n\})$. We will often make use of the simple observation that $u := \operatorname{lca}_T(x, y)$ and $v := \operatorname{lca}_T(x, z)$ for $x, y, z \in V$ are both ancestors of x and lie on the (unique) path from x to the root ρ_T. Therefore, u and v are always comparable w.r.t. $\preceq_T$.

2.3.3 *Restrictions and Refinements*

An edge $e = uv$ in a tree T is *contracted* by removing e and identifying u and v. We denote by T_e the tree that is obtained from T by contraction of the edge $e \in E(T)$, and by T_A the tree that is obtained by contraction of all edges in $A \subseteq E(T)$ (in an arbitrary order). We will usually only consider the contraction of inner edges. The contraction of inner edges in a phylogenetic T again yields a phylogenetic tree, and has an interpretation in terms of the set of clusters $\mathcal{C}(T)$ in such a way that contraction of uv (with $v \preceq_T u$) corresponds to removal of the set $L(T(v))$ from $\mathcal{C}(T)$. Together with the observation that there is at most one edge $u'v$ such that $v \preceq_T u'$ and the one-to-one correspondence with hierarchies, this implies that T_A is well-defined, i.e., the resulting tree T_A is independent of the contraction order.

Following e.g. [37, 271], we denote by $T_{|L'}$ the *restriction* of T to a subset $L' \subseteq L(T)$. More precisely, $T_{|L'}$ is obtained by identifying the (unique) minimal subtree of T that connects all leaves in L', and suppressing all vertices in that subtree that have only a single child left, i.e., by the stepwise contraction of one incident edge of such vertices until none of them remain. In particular, as a consequence of its construction, $T_{|L'}$ is again phylogenetic. The root of $T_{|L'}$ is given by $\operatorname{lca}_T(L')$. An example for this construction is given in Fig. 1. The restriction $T_{|L'}$ can alternatively be defined as the unique tree corresponding to the hierarchy $\{C \cap L' \mid C \in \mathcal{C}(T) \text{ and } C \cap L' \neq \emptyset\}$ [e.g. 114].

We say that T *displays* a tree T' on $L' \subseteq L(T)$, in symbols $T' \leq T$, if T' can be obtained from the restriction $T_{|L'}$ of T by a series of inner edge contractions. We write $T' < T$ for $T' \leq T$ and $T' \neq T$. Moreover, if $T' \leq T$ and $L(T') = L(T)$, then T is a *refinement* of T', in which case $\mathcal{C}(T') \subseteq \mathcal{C}(T)$. We say that a leaf-colored tree (T', σ') is displayed by (T, σ) if $T' \leq T$ and $\sigma(v) = \sigma'(v)$ for all $v \in L(T')$.

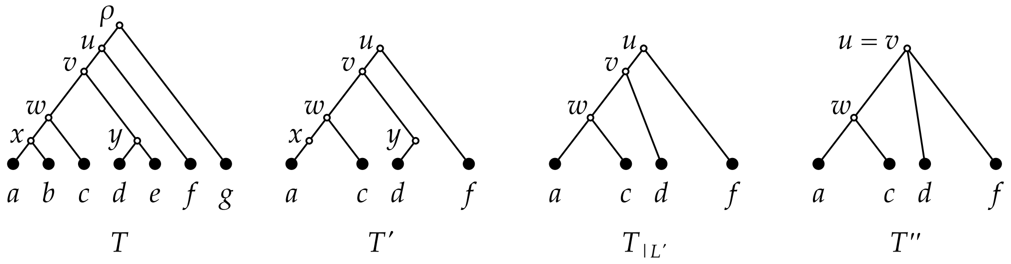


Fig. 1. The tree T has leaf set $L := \{a, b, c, d, e, f, g\}$. The minimal subtree connecting all leaves in $L' := \{a, c, d, f\}$ is T' with root u. Note that T' is not phylogenetic. The restriction $T_{|L'}$ is obtained from T by first identifying T' and then suppressing the two single-child vertices x and y. T'' is displayed by T since it can be obtained from $T_{|L'}$ by contraction of the edge uv.

2.3.4 *Rooted Triples and Triple Consistency*

A *(rooted) triple* is a binary rooted tree on three leaves, and thus, with exactly two inner vertices. We write $xy|z$ for the triple t on the leaves x, y and z if the path from x to y does not intersect the path from z to the root, i.e., if $\mathrm{lca}_t(x, y) \prec_t \mathrm{lca}_t(x, z) = \mathrm{lca}_t(y, z) = \rho_t$. In particular, a tree T with $x, y, z \in L(T)$ displays $xy|z$ if and only if $\mathrm{lca}_T(x, y) \prec_T \mathrm{lca}_T(x, z) = \mathrm{lca}_T(y, z)$. Note that $xy|z = yx|z$. Sets of rooted triples will usually be denoted by $\mathcal{R}$. We define $L_{\mathcal{R}} := \bigcup_{t \in \mathcal{R}} L(t)$ as the set of leaves appearing in $\mathcal{R}$, and $\mathcal{R}_{|L'} := \{xy|z \in \mathcal{R} : x, y, z \in L'\}$ as the restriction of a triple set $\mathcal{R}$ to a set L' of leaves. Moreover, we denote by $r(T)$ the set of all triples that are displayed by a tree T.

A set $\mathcal{R}$ of triples is *consistent* if there is a tree T (on $L_{\mathcal{R}}$) that displays every triple in $\mathcal{R}$, i.e., $\mathcal{R} \subseteq r(T)$. It can be decided in polynomial time whether a set of triples $\mathcal{R}$ is consistent using the well-known algorithm called `BUILD` by Aho et al. [7] which, moreover, constructs such a tree in the affirmative case. The `BUILD` algorithm (for the special case of triple sets as input) will play a central role in this work. The special tree that it constructs for a consistent set of triples $\mathcal{R}$ and leaf set L with $L_{\mathcal{R}} \subseteq L$ will be called the *Aho tree* and denoted by $\mathrm{Aho}(\mathcal{R}, L)$. The algorithm relies on the construction of an (undirected) auxiliary graph:

Definition 2.2. *Given a set of rooted triples $\mathcal{R}$ and a set (of leaves) L, the undirected* Aho graph, *denoted by $[\mathcal{R}, L]$, has vertex set L and edges xy if and only if $xy|z \in \mathcal{R}$ for some $z \in L$.*

Moreover, its correctness is a consequence of

Proposition 2.1. [37] *A triple set $\mathcal{R}$ on L is consistent if and only if $[\mathcal{R}_{|L'}, L']$ is disconnected for every subset $L' \subseteq L$ with $|L'| > 1$.*

The `BUILD` algorithm operates in a top-down recursive fashion in such way that, given $\mathcal{R}$ is consistent, every recursion step corresponds to a vertex in the resulting Aho tree $T := \mathrm{Aho}(\mathcal{R}, L)$. In particular, the root ρ_T is initialized in the top-level recursion step (with Aho graph $[\mathcal{R}, L]$). The tree T is constructed

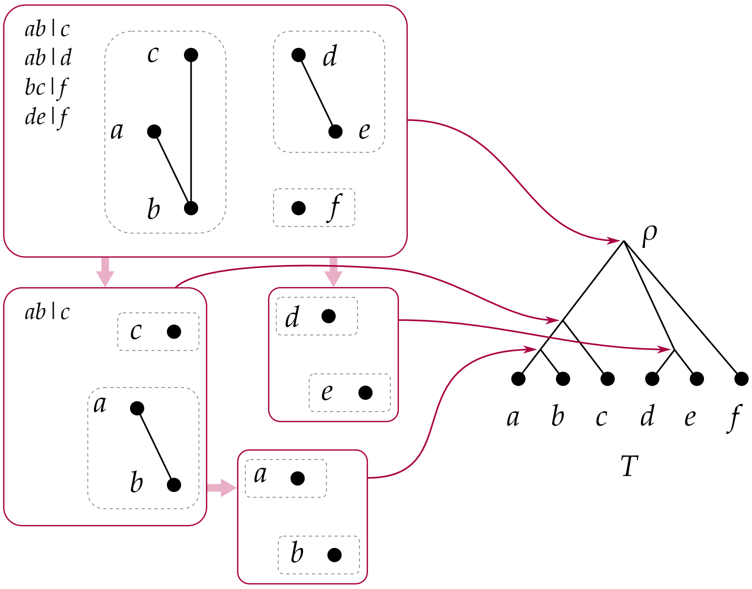


Fig. 2. Illustration of the `BUILD` algorithm with input $\mathcal{R} = \{ab|c, ab|d, bc|f, de|f\}$ and $L = \{a, b, c, d, e, f\}$. Each recursion step (pink boxes) corresponds to a vertex of the resulting tree $T = \mathrm{Aho}(\mathcal{R}, L)$ (trivial steps on single vertices are omitted in the drawing). The algorithm recurses on the connected components (gray dashed boxes) of the Aho graphs and the corresponding subsets of triples.

by recursively attaching to the root the Aho trees returned by recursive calls of `BUILD` with the vertex sets L' of the connected components of the Aho graphs (and the restricted triple sets $\mathcal{R}_{|L'}$) as input. In particular, there is a one-to-one correspondence between the children v of the root ρ', created in the step on L', and the connected components C of $[\mathcal{R}_{|L'}, L']$ given by $L(T(v)) = V(C)$. The trivial cases of the recursion are those steps where $|L'| = 1$, in which case a (sub)tree consisting of the single element in L' is returned. The algorithm fails if and only if $|L'| > 1$ and $[\mathcal{R}_{|L'}, L']$ is connected in some recursion step. By Prop. 2.1, this implies that $\mathcal{R}$ is not consistent. An example for the recursive procedure with a consistent triple set as input is shown in Fig. 2.

Since the decomposition of the Aho graphs into their connected components is unique, the Aho tree is also uniquely defined. Moreover, it is least resolved in the following sense:

Proposition 2.2. [270, Prop. 4.1] *Let $\mathcal{R}$ be a consistent set of rooted triples. Then* $\mathrm{Aho}(\mathcal{R}, L_{\mathcal{R}})$ *is a minimal rooted phylogenetic tree consistent with $\mathcal{R}$, i.e., there is no tree T' with $T' < T$ that displays all triples in $\mathcal{R}$.*

However, $\mathrm{Aho}(\mathcal{R}, L_{\mathcal{R}})$ is in general not the unique minimal tree in that sense, nor the tree with a minimal number of inner vertices [159].

Aho et al.'s [7] original motivation for the `BUILD` algorithm was an application to relational databases. In the early 1990s – about a decade later – Steel [299] was among the first to realize that it can be used to solve the *supertree problem*. The supertree problem, as it is e.g. defined in [71, 271, 299], is a generalization of the triple consistency problem. It takes a so-called profile $\mathcal{P} = \{T_1, \dots, T_k\}$, i.e., a set of k rooted trees, as input, and asks whether there is a tree T^* with leaf set $L_{\mathcal{P}} = \bigcup_{i=1}^{k} L(T_i)$ that displays all trees in $\mathcal{P}$. `BUILD` can be used to solve this problem as a direct consequence of

Proposition 2.3. [37, Thm. 1] *Let T and T' be rooted phylogenetic trees. Then $T' \leq T$ if and only if $r(T') \subseteq r(T)$ and $L(T') \subseteq L(T)$.*

Hence, the trees in a profile $\mathcal{P}$ can be encoded by the union $\mathcal{R}_{\mathcal{P}}$ of their triple sets. Henzinger et al. [139] showed that consistency of $\mathcal{R}_{\mathcal{P}}$ can be checked in $O(|\mathcal{R}_{\mathcal{P}}| \log^3 |L_{\mathcal{P}}|)$, which can be further reduced to $O(|\mathcal{R}_{\mathcal{P}}| \log^2 |\mathcal{R}_{\mathcal{P}}|)$ using the HDT data structure described by Holm et al. [146] as pointed out in [71]. Deng and Fernández-Baca [71] also present an alternative approach that avoids considering triples and Aho graphs explicitly, and runs in $O(M_{\mathcal{P}} \log^2 M_{\mathcal{P}})$ where $M_{\mathcal{P}}$ is the total number of vertices and edges in $\mathcal{P}$. We note in passing that the corresponding consistency check for unrooted trees is NP-complete [299]. In practice, efficient supertree methods – in particular, such that can handle inconsistencies – have become relevant in the endeavor to summarize published species trees in a comprehensive tree of life, see e.g. [142].

We will also need to consider trees that explicitly do not display certain triples. To this end, we generalize the notion of consistency for pairs of triple sets:

Definition 2.3. *A pair of triple sets $(\mathcal{R}, \mathcal{F})$ is* consistent *if there is a tree T that displays all triples in $\mathcal{R}$ but none of the triples in $\mathcal{F}$. In this case, we say that T* agrees with *$(\mathcal{R}, \mathcal{F})$.*

For $\mathcal{F} = \emptyset$, this definition reduces to the usual notion of consistency of $\mathcal{R}$. Consistency of $(\mathcal{R}, \mathcal{F})$ can also be checked in polynomial time. The algorithm `MTT`, named for *mixed triplets problem restricted to trees*, constructs a tree T that agrees with $(\mathcal{R}, \mathcal{F})$ or determines that no such tree exists [128]. It can be seen as a generalization of `BUILD`. Each recursion step of `MTT` starts by initializing an auxiliary partition $\mathcal{D}$ of L' according to the connected components of the Aho graph $[\mathcal{R}_{|L'}, L']$. In a second step, it stepwise merges distinct sets D and D' in $\mathcal{D}$ if there are triples $xy|z \in \mathcal{F}_{|L'}$ with $x, y \in D$ and $z \in D'$ until no such triples remain. The algorithm then recurses on the sets in the so-constructed partition $\mathcal{D}$, and fails if and only if $|L'| > 1$ and $|\mathcal{D}| = 1$ at some point.

Theorem 2.1. [128, Thm. 1] *Algorithm `MTT` outputs a phylogenetic tree T that agrees with the pair $(\mathcal{R}, \mathcal{F})$ (defined on n distinct leaves) if and only if $(\mathcal{R}, \mathcal{F})$ is consistent in $O(|\mathcal{R}| \cdot n + |\mathcal{F}| \cdot n \log n + n^2 \log n)$ time.*

Moreover, any pair of triples $(\mathcal{R}', \mathcal{F}')$ such that $\mathcal{R}' \subseteq \mathcal{R}$ and $\mathcal{F}' \subseteq \mathcal{F}$ for a consistent pair $(\mathcal{R}, \mathcal{F})$ remains consistent since any tree that agrees with $(\mathcal{R}, \mathcal{F})$ clearly displays all triples in $\mathcal{R}'$ and none of the triples in $\mathcal{F}'$. Hence, we have

Observation 2.2. *Let $\mathcal{R}' \subseteq \mathcal{R}$ and $\mathcal{F}' \subseteq \mathcal{F}$ for a consistent pair of triple sets $(\mathcal{R}, \mathcal{F})$. Then $(\mathcal{R}', \mathcal{F}')$ is consistent.*

Following Bryant and Steel [37], an inner edge e of a rooted tree T is *distinguished* by a triple $ab|c \in r(T)$ if the path from a to c in T intersects the path from b to the root ρ_T precisely on the edge e. In other words, $e = uv$ with $v \prec_T u$ is distinguished by $ab|c$ if $\operatorname{lca}_T(a,b) = v$ and $\operatorname{lca}_T(a,b,c) = u$.

A set of triples $\mathcal{R}$ on a set of leaves L is *strictly dense* if, for all three distinct $x, y, z \in L$, exactly one of the triples $xy|z$, $xz|y$, and $yz|x$ is contained in $\mathcal{R}$. For later reference, we will need

Lemma 2.1. [133, SI Appendix, Lemma 7] *Let $\mathcal{R}$ be a consistent set of triples on L. Then there is a strictly dense consistent triple set $\mathcal{R}'$ on L such that $\mathcal{R} \subseteq \mathcal{R}'$.*

Lemma 2.1 can be seen by considering a binary refinement T' of an arbitrary tree T that displays a consistent set $\mathcal{R}$. Clearly, T' also displays $\mathcal{R}$, i.e., $\mathcal{R} \subseteq r(T')$. Moreover, we have that $r(T')$ is strictly dense as a consequence of T' being binary.

2.3.5 *The Span and the Closure Operator*

Following [37, 269], we call the *span* of $\mathcal{R}$, denoted by $\langle \mathcal{R} \rangle$, the set of all trees with leaf set $L_{\mathcal{R}}$ that display $\mathcal{R}$. With this notion, we define the *closure operator* for consistent triple sets by

$$\operatorname{cl}(\mathcal{R}) = \bigcap_{T \in \langle \mathcal{R} \rangle} r(T), \tag{1}$$

i.e., a triple t is contained in $\operatorname{cl}(\mathcal{R})$ if all trees that display $\mathcal{R}$ also display t. In particular, $\operatorname{cl}(\mathcal{R})$ is again consistent. The map cl is a closure in the usual sense on the set of consistent triple sets, i.e., it is extensive $[\mathcal{R} \subseteq \operatorname{cl}(\mathcal{R})]$, monotonic $[\mathcal{R}' \subseteq \mathcal{R} \implies \operatorname{cl}(\mathcal{R}') \subseteq \operatorname{cl}(\mathcal{R})]$, and idempotent $[\operatorname{cl}(\mathcal{R}) = \operatorname{cl}(\operatorname{cl}(\mathcal{R}))]$ [37, Prop. 4]. A consistent set of triples $\mathcal{R}$ is *closed* if $\mathcal{R} = \operatorname{cl}(\mathcal{R})$.

The following characterization of triples that are contained in the closure relies on Aho graphs:

Proposition 2.4. [35, Cor. 3.9] *Let $\mathcal{R}$ be a consistent set. Then $ab|c \in \operatorname{cl}(\mathcal{R})$ if and only if there is a subset $L' \subseteq L_{\mathcal{R}}$ such that the Aho graph $[\mathcal{R}_{|L'}, L']$ has exactly two connected components, one containing both a and b, and the other containing c.*

Following [101, 114], a set of rooted triples $\mathcal{R}$ *identifies* a tree T on L if T displays $\mathcal{R}$ and every other tree on L that displays $\mathcal{R}$ is a refinement of T.

Proposition 2.5. [114, Lemma 2.1] *Let T be a phylogenetic tree and $\mathcal{R} \subseteq r(T)$. Then $\operatorname{cl}(\mathcal{R}) = r(T)$ if and only if $\mathcal{R}$ identifies T.*

2.3.6 *Special Trees*

A *star (tree)* is a tree T that has exactly one inner vertex ρ_T, i.e., all of its leaves are children of the root ρ_T. A *caterpillar (tree)* is a binary tree in which every inner vertex has at most one child that is an inner vertex.

Following e.g. [103], a tree T is a *planted phylogenetic tree* if (i) it has a root 0_T that has a single child $\rho_T = \operatorname{lca}_T(L(T))$, and (ii) the subtree $T(\rho_T)$ is a phylogenetic tree in the usual sense, i.e., as introduced above. We will call 0_T the *planted root*, as opposed to the *conventional root* ρ_T. Planted trees are useful in phylogenetics (where T represents the history of a group of genes or species) to model events that took place before the first branching event (represented by ρ_T). In particular, the unique *planted edge* $0_T\rho_T$ represents the "ancestral lineage". Note that by definition $0_T \notin L(T)$. Moreover, we set $V^0(T) := V(T) \setminus (L(T) \cup \{0_T\})$ for planted trees, i.e., 0_T is also not an inner vertex. We will make extensive use of planted trees in Chapters 6 to 8. Therein, whenever there is no danger of confusion, we will refer to both rooted phylogenetic trees and planted phylogenetic trees simply as trees.

2.4 COGRAPHS & COTREES

Cographs are a well-studied class of undirected graphs that, in particular, appear often in the mathematical modeling of the relationship of genes [103, 132, 185]. Cographs are generated from single-vertex graphs by repeated application of disjoint union and join operations:

Definition 2.4. *An undirected graph G is a cograph if*

1. *G is a K_1,*
2. *$G = H \mathbin{\dot\cup} H'$ is the disjoint union of two cographs H and H', or*
3. *$G = H \triangledown H'$ is the join of two cographs H and H'.*

The recursive construction defines a rooted tree (T,t) with $t\colon V^0(T) \to \{0,1\}$, called *cotree*. Its leaves are the vertices of the cograph G, i.e., the K_1s, while its inner vertices u represent the disjoint union or join operations, labeled as $t(u) = 0$ and $t(u) = 1$, respectively. Hence, for a given cograph G and its cotree (T,t), we have $xy \in E(G)$ if and only if $t(\mathrm{lca}_T(x,y)) = 1$. Since both the disjoint union and the join are associative operations, a cotree is not necessarily binary. In general, therefore, the cograph $G = G[\rho_T]$ can be obtained from a cotree (T,t) using

$$G[u] = \begin{cases} \dot{\bigcup}_{v \in \mathsf{child}_T(u)} G[v] & \text{if } t(u) = 0 \\ \bigtriangledown_{v \in \mathsf{child}_T(u)} G[v] & \text{if } t(u) = 1 \\ (\{u\}, \emptyset) & \text{if } u \text{ is a leaf} \end{cases} \tag{2}$$

where $G[w]$ with $w \in V(T)$ denotes the cograph corresponding to the subtree $(T(w), t_{|L(T(w))})$.

Contraction of all tree edges $uv \in E(T)$ with $t(u) = t(v)$ results in the *discriminating cotree* $(T_G, \hat{t})$ of G with cotree-labeling $\hat{t}$ such that $\hat{t}(u) \neq \hat{t}(v)$ for any two adjacent interior vertices of T_G. The discriminating cotree $(T_G, \hat{t})$ is uniquely determined by G [57]. Cographs have a large number of equivalent characterizations. In this work, we will need the following classical results:

Proposition 2.6. [57] *Given an undirected graph G, the following statements are equivalent:*

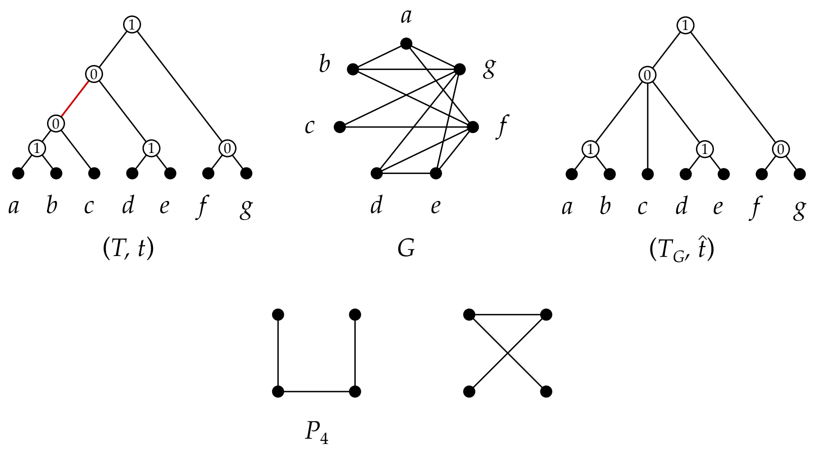


Fig. 3. Top row: Example for a cograph G and a corresponding cotree (T, t). The unique discriminating cotree $(T_G, \hat{t})$ is obtained from (T, t) by contraction of the edge that is highlighted in red. Bottom row: The P_4 is the characteristic forbidden induced subgraph of cographs (cf. Prop. 2.6). Its complement (drawn on the r.h.s.) is again a P_4.

1. *G is a cograph.*
2. *G does not contain a P_4, i.e., a path on four vertices, as an induced subgraph.*
3. $\mathrm{diam}(H) \leq 2$ *for all connected induced subgraphs H of G.*
4. *Every induced subgraph H of G is a cograph.*

In particular, Prop. 2.6(4) states that cographs are a so-called *hereditary* class of graphs. Moreover, they coincide with the P_4-free graphs, i.e., graphs that do not contain a P_4 as an induced subgraph. A P_4 is the smallest example of a connected graph whose complement is again connected (cf. Fig. 3). Therefore, graphs containing an induced P_4 cannot be decomposed entirely into single vertices by a recursive application of the complement operation on its connected components. As a consequence of their construction (cf. Def. 2.4), this is possible for cographs. In fact, the term *cograph* is short for *complement-reducible graph* [57].

2.5 COMPUTATIONAL COMPLEXITY

The *complexity* of an algorithm measures the amount of time or resources that it requires. We will mainly consider the (worst-case) *time complexity* which measures the number of elementary operations, which themselves are assumed to take constant time, that are executed during the computation on a (usually idealized) machine.

The time complexity of an algorithm is typically expressed as a function of the input size using the so-called *big O notation*. We say that a function $f(n)$ is $O(g(n))$ if there are constants c and n_0 such that $f(n) \leq c \cdot g(n)$ for all $n > n_0$. Thus, with n denoting the input size, an algorithm runs in $O(g(n))$

(time) if the number of required elementary operations as a function of n is $O(g(n))$. If g is a polynomial of finite degree, then the algorithm is said to run in *polynomial time* and called a *polynomial-time algorithm*. Common complexity classes of algorithms are constant ($O(1)$), logarithmic ($O(\log n)$), linear ($O(n)$), quadratic ($O(n^2)$), and exponential (e.g. $O(2^n)$) time algorithms. In this work, the input size will usually be measured in terms of the number of vertices and/or edges of a graph, or the number of leaves of a tree. When speaking of a linear-time algorithm that takes a graph $G = (V, E)$ as input, we usually mean $O(|V| + |E|)$, i.e., the algorithm runs in linear time w.r.t. the number of vertices and edges.

As a consequence of their definition as upper bounds of the asymptotic behavior of algorithms, complexity classes can be ordered according to the inclusion of smaller classes, e.g., all algorithms that run in $O(n)$ also run in $O(n^2)$, the converse of which is not true. We are usually interested in finding the smallest complexity class of an algorithm in that sense. Correspondingly, the complexity of a problem is defined as the smallest complexity of all algorithms that solve it. In general, determining the complexity of a problem is a difficult task since one has to take all possible algorithms into account. Nevertheless, it is often possible to classify problems according to their difficulty. In the following, we will introduce some important concepts and methods that are used in this field.

In complexity theory, *decision problems* are such problems for which the problem instances can either be answered with “yes” or “no”. Even though other problem classes such as *optimization* and *search problems* are often of higher practical interest, decision problems are usually the starting point for studying the complexity of a problem. In particular, other types can often be transformed into decision problems. In this work, we will e.g. ask what the minimum number of edges/arcs is that need to be inserted into or deleted from a graph to obtain a graph with a certain property – a classical optimization problem. One can alternatively ask whether there is a set of edge/arc insertions/deletions of cardinality at most k such that the modified graph satisfies the desired property, and then stepwise increase or decrease k. Hence, we have obtained possibly multiple decision problems from the original optimization problem. Accordingly, we will frequently speak of the optimization and the decision version of such problems.

The class of all decision problems that can be solved by a (not necessarily known) polynomial-time algorithm is denoted by P. The complexity class P is contained in the class NP, which comprises all decision problems for which a given solution can be verified as being valid or not in polynomial time. NP stands for “non-deterministic polynomial time” which refers to the fact that NP problems are solvable in polynomial time on a so-called nondeterministic Turing machine, a theoretical model of a machine that is able to follow all possible paths of a computation at the same time, and therefore, finds and verifies a solution, if one exists, in polynomial time. A problem is called *NP-hard* if it is at least as hard as the hardest problems in NP, and *NP-complete* if it is additionally contained in NP. Well-known examples of NP-complete problems are the SATISFIABILITY (SAT) problem, the CHROMATIC NUMBER of graphs, and 0-1 Integer Linear Programming (ILP) [165]. Several hundreds more have been collected by Garey and Johnson [99].

There are thousands of problems in NP, including all NP-complete problems, for which no one has found a polynomial-time algorithm [19], leading to the widely accepted conclusion that P$\neq$NP. So far, however, this has not been proven. In fact, the *P vs NP problem* is one of the biggest unsolved problems in computer science and among the seven Millennium Prize Problems [44]. At the same time, it is reasonable to believe that a nondeterministic Turing machine cannot be realized in practice. Even quantum computers are probably unable to solve NP-complete problems efficiently [2, 26]. The relationship of NP and BQP, the class of decision problems that can be solved in polynomial time on a quantum computer, has been studied by computer scientists since before the first notable realizations of quantum computers and largely remains an open problem [1, 227, 247]. The latter arguments together with the fact that non-polynomial-time algorithms are usually workable only for small instances in a reasonable amount of time suggest that there are problems for which an exact solution cannot be found efficiently, i.e., in polynomial time. In particular, this provides a justification for the development of faster *heuristic* algorithms that produce “good” but not necessarily optimal solutions.

An important tool to study the complexity of (decision) problems are *reductions*. A reduction is an algorithm that transforms instances of problem A into instances of another problem B in a way that allows the reconstruction of a solution for A from a solution for B. An important condition for a reduction to be useful is that it is easy w.r.t. the studied complexity class. Therefore, polynomial-time reductions are employed to prove the membership of a problem to the NP-hard problems. If problem A can be reduced to problem B using a polynomial-time reduction and problem B can be solved in polynomial time, then problem A clearly also can be solved in polynomial time. If, on the other hand, we assume that problem A is not solvable in polynomial time, then we obtain a contradiction and can conclude that problem B also cannot be solved in polynomial time.

Integer Linear Programming (ILP) is a useful tool to derive exact solutions for NP-complete problems by means of polynomial-time reductions. An integer linear program consists of

(i) a set of integer variables $x_1, \ldots, x_n$,

(ii) a set of linear constraints of the form $\sum_{1 \leq j \leq n} a_{ij} x_j \leq b_i$ with $a_{ij}, b_i \in \mathbb{R}$, and

(iii) a linear objective function of the form min / max $\sum_{1 \leq j \leq n} c_j x_j$ with $c_j \in \mathbb{R}$.

In this work, we will, in particular, make use of ILPs with binary variables only, i.e., $x_j \in \{0, 1\}$ for all $1 \leq j \leq n$, to formulate certain graph modification problems in Sec. 5.4. Even though ILP is NP-hard, there exist solvers such as IBM ILOG CPLEX™ Optimizer [154] and Gurobi Optimizer [118] that allow the computation of exact solutions at least for moderate-size instances.

3

CONCEPTS AND METHODS IN PHYLOGENETICS

PHYLOGENETICS is a field in biology that addresses the reconstruction of evolutionary histories and the relationship among biological entities such as species (or higher-order taxa) and genes. The aim is usually to construct a *phylogenetic tree* or *phylogeny* that describes the evolution from a common ancestor (the root) to the extant species or genes (the leaves) that we can observe now through a series of branching events (the inner vertices). Rooted and unrooted trees as well as their application as models in biology have been studied extensively, see e.g. [88] and [271] for standard textbooks on the topic.

There are a number of evolutionary processes that trees cannot accommodate. Events such as recombination in (parts of) genes, hybridization, i.e., interbreeding between different species, and horizontal gene transfer can result in a rather reticulate evolutionary history of a group of species and genes. Therefore, phylogenetic networks as a more general framework that enable both bi- or multifurcations (as in trees) and reticulations have received great attention especially in the last two decades, see e.g. [153] and [217] for reviews. At the same time, software to infer and analyze such networks has been developed [e.g 36, 286, 328]. However, despite the fact that they may not cover all subtleties of evolutionary histories, trees are still the most commonly used model in phylogenetics [241]. In particular, it will also be assumed throughout this work that evolution progresses in a tree-like manner.

3.1 EVOLUTIONARY SCENARIOS AND THE TYPES OF HOMOLOGY

In a species tree, usually denoted by S, the inner vertices represent the divergence of some species into two or more descendant species. Even though speciation is, as we will discuss briefly below, a gradual process, we will assume that it takes a sufficiently small period of time such that a single point in time can be assigned to the speciation vertices. In modern (molecular) phylogenetics, species are represented by their genomes, i.e., the heritable information in form of a sequence of nucleic acids in the DNA (or RNA in the case of some groups of viruses).

The genome of an organism comprises the genes as well as intergenic regions. Genes have been defined in various different ways in history and also in the more recent literature [105, 292]. For example, Gerstein et al. [105] define a gene as "a union of genomic sequences encoding a coherent set of potentially overlapping functional products", where functional products are either proteins or functional RNA molecules. Here, it will be convenient to put a larger emphasis on the association of a gene to a sequence of nucleic or amino acids, and to use the term *gene* in a more general way to refer to any kind of heritable sequence information.

Similar to species trees, a gene tree (usually denoted by T) describes the evolutionary history of a gene family, i.e., a group of genes that share a common ancestry. Genes that have evolved from a common ancestor usually exhibit a certain degree of sequence similarity and have been termed *homologous* or *homologs* by Fitch [89], as opposed to *analogous* genes which fulfill similar functions but do not share a common ancestry. This terminology parallels the notion of homologies and analogies used for morphological traits of organisms [43]. For example, the wings of birds and butterflies are analogous traits since they evolved independently on separate branches of the bilaterian phylogeny, and are the result of so-called convergent evolution. On the other hand, the four limbs of tetrapods are homologous as they emerged from a common ancestry – as different as for example a human arm and a whale's front flipper may appear. Similarly, homologous genes do not necessarily have the same function. In practice, it is usually known in which genome/species an extant gene resides. We will encode this knowledge in form of a leaf-coloring $\sigma\colon L(T) \to M$, which assigns to each leaf of T a species from a set M, and often also call the pair (T, σ) a gene tree.

As a consequence of their association with a genome, the evolution of a gene family is closely tied to the evolution of the underlying organisms. In particular, the descendant species of a speciation event each obtain copies of the genes in the parent species which, from then on, continue to evolve independently, at least in the absence of events such as horizontal gene transfer and hybridization. Therefore, the divergence of species also imposes a divergence on the genes. Conversely, the evolution of gene families is additionally driven by events that are independent of branching events in the species phylogeny. The most important of them and the ones that will be considered in this work are the following:

1. *(gene) duplication* – a genomic sequence containing one or more genes gets copied,
2. *(gene) loss* – a genomic sequence containing one or more genes gets lost by deletion or by extensive accumulation of smaller mutations, and
3. *horizontal gene transfer (HGT)* – genetic material gets transferred between organisms that are not in a parent-offspring relationship.

Some biological mechanisms and examples for these events will be discussed in Sec. 3.2. As a consequence of these three processes, the topologies of gene trees are not necessarily congruent with the topology of the underlying species tree. Another phenomenon that can produce incongruent topologies is incomplete lineage sorting (ILS) which refers to polymorphisms, i.e. the occurrence of two or more variants of a gene at the same genomic locus (alleles) within a population, that are retained over multiple speciation events followed by a (differential) loss of alleles in the descendant species. Throughout, we will not consider ILS explicitly and refer to [202, 205] for reviews on how to model and handle this phenomenon.

We will use the term *(evolutionary) scenario* to describe a pair consisting of a gene tree and a corresponding species tree together with some form of relationship between the two. Despite their possible incongruence, species and gene trees constrain one another in the sense that the gene tree is "embedded" into the species tree in a manner that is, in particular, time-consistent [81, 228]. This means that it must be possible to assign a timing to each event such that ancestral events predate their descendants and, moreover, no conflicts exist between the timing of the gene tree's events and the timing of the parts of the species tree into which they are embedded. Mathematically, such embeddings are formalized by a so-called *reconciliation (map)*. In the most common setting, a reconciliation maps the vertices of the gene tree to vertices (or the union of vertices and edges) of the species tree and satisfies a set of axioms or constraints that depend on the exact purpose such as the modeling of scenarios that do or do not include HGT [103, 115, 228, 252, 316]. In Sec. 7.1, several examples of specific reconciliation maps will be discussed. Moreover, the reconciliation map for duplication-loss scenarios that first appeared in its form in [103] will be introduced formally in Chapter 6. In Chapters 7 and 8, relaxed scenarios will be introduced as a more general framework that also allows horizontal gene transfer.

A reconciliation naturally gives rise to different types of inner vertices of the gene tree. In particular, the vertices that are mapped to inner vertices of the species tree rather than an edge are called speciation vertices. In the absence of HGT, all other inner vertices must be duplication events. Using these notions, the following distinction of pairs of genes in terms of their last common ancestor can be applied to both reconciled pairs of trees and the (generally unknown) true evolution of a gene family within its underlying species phylogeny:

Definition 3.1. *Two genes are called* orthologs *(or* orthologous*) if their last common ancestor was a speciation event, and* paralogs *(or* paralogous*) if their last common ancestor was a duplication event.*

We will work with this definition of orthology and paralogy which was introduced by Walter Fitch [91] and has been adopted widely, see e.g. [11, 14, 96, 173, 226]. Other notions of orthology will be discussed briefly in Sec. 3.3.3. Note that orthology (as well as paralogy) defines a symmetric and irreflexive relation on a set of homologous genes. As a consequence, this relation can be represented by an undirected graph – the so-called orthology graph (or paralogy graph, respectively). Both relations are in general not transitive [91]. Nevertheless, Hellmuth et al. [133] showed that orthology graphs coincide with the well-known class of cographs (see also Thm. 6.1). To see this, observe that we can obtain a cotree (T, t) for an orthology graph G on the leaf set of a gene tree T by defining the labeling $t(u) = 1$ if and only if u is a speciation vertex for all $u \in V^0(T)$. The relationship of a cotree and its cograph together with Def. 3.1 almost immediately implies that G must be a cograph.

In order to account for horizontal gene transfer, we introduce a third relation which was also introduced by Fitch [91]:

Definition 3.2. *Two genes are called* xenologs *(or* xenologous*) if the history, since their last common ancestor, involves an interspecies horizontal transfer of the genetic material (i.e. an HGT) for at least one of them.*

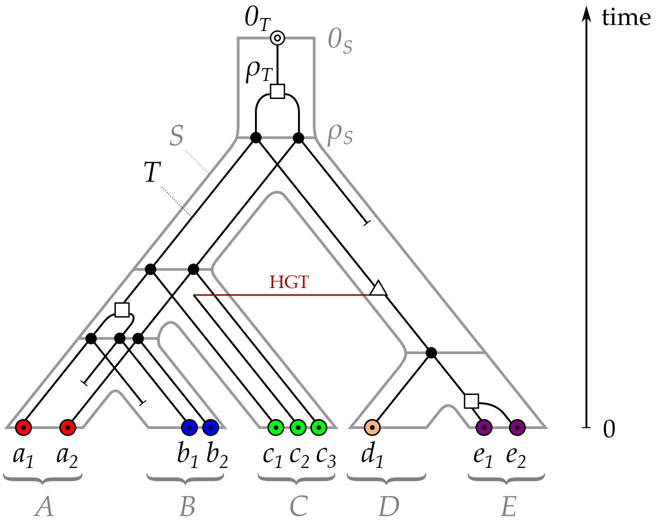


Fig. 4. An evolutionary scenario consists of a (leaf-colored) gene tree (T, σ) embedded into a species phylogeny S (in this example, on a set of five extant species/genomes $\{A, B, C, D, E\}$) which is drawn in a tube-like fashion with horizontal lines corresponding to the vertices. The coloring σ of the (non-loss) leaves of the gene tree represents the (extant) species in which the (extant) genes reside. Inner nodes of the gene tree correspond to gene duplications (□), horizontal gene transfer events (△), or speciation events (●), the latter coincide with the inner nodes of the species tree S. For completeness, the branches leading to losses (⊣) are also shown. The reconciliation is given implicitly by the drawing of T within S. Similarly, a (relative) timing of the events is given. If a timing is considered explicitly, we will usually set the timing of the extant species/genes to zero, whereas ancestors receive larger time stamps. We will often omit the drawing of the time scale. Adapted from [258].

Note that, in contrast to orthology and paralogy, xenology is not defined in terms of the type of the last common ancestor of two genes. In particular, an edge separating two genes in the tree that corresponds to an HGT event is not necessarily incident to their last common ancestor, and thus, the two genes can be both orthologs (or paralogs) and xenologs. Despite the asymmetric nature of horizontal gene transfer events – from a donor to a recipient species, we consider xenology as an undirected relationship here (see also [135]) rather than the directed version that was e.g. investigated in [101, 134]. As we will see in Chapter 8, it is easier to infer undirected xenology information from the kind of data that we consider here.

Fig. 4 shows an example for an evolutionary scenario whose gene tree includes all four event types: speciation events, gene duplications, a horizontal gene transfer, and losses. Therein, the gene pairs (a_1, c_1) and (b_1, e_1) are examples for orthologs since, in both cases, the last common ancestor is a speciation event. In contrast, (a_1, b_1) or (a_1, c_3) have duplication events as last common ancestors, and thus, they are paralogs. The gene c_2 emerged from its parent by a horizontal gene transfer. Therefore, it is xenologous to any other gene. For completeness, we will often include branches leading to losses in the drawings of evolutionary scenarios. *However, we will assume that a gene tree does not contain leaves that correspond to losses if not explicitly stated otherwise.*

Speciation

Similar to the concept of genes, the definition of the term *species* is an ongoing process of controversies that required several re-interpretations in the light of new findings [3, 206]. While Darwin viewed the variety of organisms as a continuum in which species are only distinguishable by gaps in the distribution of morphological traits [cf. 206], Ernst Mayr includes reproductive barriers in the so-called biological species concept by defining species as "groups of actually or potentially interbreeding natural populations, which are reproductively isolated from other such groups" [208]. This definition has been widely adopted in biology textbooks [e.g. 43], however, it fails in doing justice to organisms that (predominantly) reproduce asexually or to clades in which hybridization is common as it is e.g. the case for certain taxa of plants [10, 111].

One commonly distinguishes between different modes of speciation [40, 277]: Allopatric speciation refers to the geographical separation of populations, e.g. resulting from continental drift or mountain formation, which stops the exchange of genetic material between them. The following independent evolution eventually leads to a reproductive incompatibility by genetic drift and the accumulation of mutations. Sympatric speciation, on the other hand, refers to the divergence of species in the absence of such extrinsic barriers. Mechanisms that drive in particular the latter kind of speciation include the adaptation of parts of a population to certain ecological parameters (so-called niches), genetic drift (a change of the frequency of certain genetic variants resulting from random sampling), and the acquisition of new genes and features resulting e.g. from (non-)homologous recombination, horizontal gene transfer, or hybridization [276, 277].

Gene Duplications

Gene duplication provides new genetic material to be subject to mutation and evolution, and thus, for the establishment of new features. A large and well-known group of genes that most likely emerged from extensive tandem duplication are the homeobox (Hox) genes, which are crucial in the regulation of the formation of body segments in bilaterian embryonic development [145, 230]. This example demonstrates that gene duplication has been a major driving force e.g. in the development of the morphological diversity among species.

Genomic sequences containing parts of a gene, a whole or even multiple genes can be duplicated by a variety of mechanisms with different characteristics, see e.g. [203, 240, 341] for reviews. So-called tandem duplications are the result of unequal crossing-over events, i.e., the misalignment of two chromosomes during meiosis leading to the exchange of unequally long regions between them. In contrast to tandem duplications, segmental duplications produce homologous sequences that are not necessarily neighboring in the genome. They often result from non-allelic homologous recombination [22, 279]. Gene duplicates in a more or less random position of the genome can be generated by a process called retroposition in which processed RNA is reverse-transcribed into DNA and then inserted into the genome [34]. As a consequence, the introns, i.e., parts of the transcribed sequence that are spliced out during the RNA processing, are missing in the copy, which is indicative of this type and makes original and copy distinguishable. A fourth important mechanism is whole-genome duplication (WGD), or polyploidization. Albeit observed in many branches of eukaryotes, this phenomenon is particularly common in plants [234] and has been of key importance for the development of innovations in angiosperms [240, 287].

Once a duplicate is introduced into the genome, its fate may simply be to accumulate mutations and eventually lose its ability to be described, and thus, become nonfunctional. This process is called pseudogenization. In fact, this seems to be the most common case since, like most mutations, duplications often have a deleterious effect on the fitness of individuals [199]. For the cases in which the two (or more) versions of a duplicated gene are retained, several models have been developed to make predictions about the acquisition of new functions and, in particular, the possibly unequal rates of evolution resulting from changed selective forces. Ohno's neofunctionalization model [231] predicts that one copy retains its original function, while the other becomes relieved from selective pressure which leads to the accumulation of mutations and, in some cases, to the acquisition of novel functions. The duplication-degeneration-complementation (DDC) model was proposed by Force et al. [93]. It states that both copies will lose the ability to fulfill their original function alone in the 'degeneration phase', leading to the situation that both copies must be retained and subject to selective pressure ('complementation phase'). The specialization model [72, 151] is closely related and applies to genes with multiple functions. The main idea is that multiple functions lead to a so-called adaptive conflict, i.e., the gene is prohibited from adapting to fulfill either function efficiently. Therefore, an additional copy of the gene may allow an escape from adaptive conflict (EAC). For a review on several variations and extensions of these models, it is referred to [155]. In practice, characteristics of various models have been observed [70, 72, 127, 313] suggesting that these mechanisms coexist in biology.

Gene Losses

The loss of a gene may refer to either the absence of some gene that is identified by the comparison with different genomes or the non-functionalization of a gene [8]. The latter is the consequence of a loss-of-function mutation in the coding region or in its regulatory sequence, in particular, in its promoter. A non-functional gene usually undergoes pseudogenization, i.e., it accumulates mutations, due to the missing selective pressure. Eventually, this may lead to the first situation, i.e., the actual absence of the gene. Apart from this gradual process, the second mechanism of gene loss may be a single mutational event such as unequal crossing over or the activity of a transposable element or retrovirus [8]. In this work, we will mainly refer to the actual absence of genes rather than non-functionalization when speaking about gene losses. In particular, we consider gene families which are usually determined by sequence similarity, and assume that lost genes are not observable and may only be inferred indirectly.

It seems rational to believe that gene losses should have a deleterious effect in most cases. However, mutagenesis studies intended to discover functions of genes suggest that most genes, even in eukaryotes, are dispensable at least to some extent, i.e., their knockout leaves organisms viable and, in some cases, has no apparent effect at all [177]. The 'less-is-more' hypothesis even states that gene losses are a driving force for innovations that enable organisms to improve their fitness and to adapt to new environments [232]. In particular, there is evidence that losses were of importance in human evolution [218, 233, 298]. Other notable examples for innovations associated with losses include inactivating mutations in dentin- and enamel-related genes in the common avian ancestor that were probably necessary for the development of the horny beak of modern birds [211], and the loss of hair- and epidermis-related genes in cetaceans as an adaptation of their ancestors to an aquatic environment [278].

Horizontal Gene Transfer

Horizontal gene transfer (HGT), also called lateral gene transfer, is the transmission of genetic material between organisms that are not in a parent-offspring relationship, i.e., it is decoupled from reproduction. Since the discovery of the first mechanisms for HGT in the first half of the twentieth century [e.g. 308], the role of HGT as one of the major forces that shape the genomes of bacteria, archaea, and even eukaryotes has been investigated in-depth and widely accepted, for recent reviews see e.g. [63, 291].

The probably best-known examples of HGTs are the acquisition of antibiotic resistance and pathogenicity factors in microbes posing a threat to medical systems worldwide [143, 191, 209]. However, another important aspect is the exchange of genes involved in metabolic pathways. A peculiar example is the transfer of a gene encoding a polysaccharide-digesting enzyme from a parasite of marine algae to the gut bacteria of Japanese people enabling the latter to digest the cell walls of seaweed [129]. The phenomenon is particularly frequent in prokaryotes [224, 291], with variable rates across different clades [4], but multiple cases involving eukaryotes have also been described [6, 152, 167, 192, 215, 267]. For example, it was found that the evolution of C_4-photosynthesis in plants has been mediated by HGT [54]. Moreover, the sporadic occurrence of type II antifreeze protein in three widely separated branches of teleosts most likely is the result of HGT [112].

There are three main mechanisms of HGT in prokaryotes [291]: Conjugation is the transfer of genetic material from a donor to a recipient cell via the formation of a so-called pilus that is produced by the donor cell, and thus, requires direct physical contact. Transformation is the uptake of DNA from the environment into the cell. Transduction refers to the situation in which foreign genetic material is injected into the cell by a bacteriophage, i.e., a virus that infects prokaryotes. More recently identified mechanisms include so-called gene transfer agents (GTAs) [297] and cell fusion [219]. According to the exact mechanism of the integration of foreign DNA into the host genome (or its co-existence with the latter e.g. in form of a so-called plasmid), HGT may be additive, in which case its effect is similar to gene duplications, or lead to the replacement of a vertically inherited homolog.

3.3 INFERENCE METHODS

For the longest time in the history of taxonomy and phylogenetics so far, biologists' only tools to disentangle the diversity of life on this planet were comparisons of morphological and physiological traits of different extant organisms and, in some cases, fossils. Clearly, this kind of data is still one of the main pillars in the reconstruction of phylogenies. However, it is often insufficient for resolving the details of the evolutionary relationships. Since the sequencing of the first organisms' genomes – including the bacteriophage ϕX174 in the 1970s [255], *Haemophilus influenzae* as the first completely sequenced prokaryote in 1995 [92], and eventually, the publication of the "finished" human genome in 2004 [156] – the emergence of various high-throughput approaches to DNA sequencing have drastically decreased the amount of time and money required for whole-genome sequencing, see e.g. [248] for a review. This has led to an enormous amount of DNA sequence data, large parts of which are publicly available and at least partially annotated in databases such as GenBank [27], RefSeq [244] and Ensembl [338]. At the same time, a broad spectrum of mathematical models and computational methods have been developed for the use of this molecular data in phylogenetic analyses, see e.g. [18, 88, 335, 336] for well-known textbooks on the topic.

3.3.1 *Gene Similarity, Distances, and the Molecular Clock*

DNA is a molecule composed of two complementary polynucleotide strands. The genetic information is encoded as the specific sequence of the monomers (nucleotides) in these strands, or, more precisely, the sequence of the four nucleobases adenine (A), cytosine (C), guanine (G), and thymine (T) being the variable component of the nucleotides. Similarly, proteins are comprised of one or multiple chains of the 20 (standard) amino acid residues. Naturally occurring errors in the replication or repair of DNA, as well as many chemicals and radiation with a wavelength shorter than light (UV and ionizing radiation), can lead to alterations of the genomic sequences in a cell, so-called mutations [9, 43]. Mutations include the substitution, insertion, or deletion of a single or a few nucleotides but also larger-scale alterations such as rearrangements of DNA segments within or among chromosomes. Since the amino acid sequence of a protein is encoded in the DNA, mutations in the latter may also affect an encoded protein resulting e.g. in the substitution of a single or multiple amino acids or a truncation of the protein.

While most mutations affect somatic cells or decrease the reproductive fitness of the offspring, some mutations become established within a population or a whole species – especially if they have an advantageous effect on the fitness. As a consequence, genes (or proteins) that share a common ancestry, i.e. homologs, usually have a similar but different nucleotide (amino acid, resp.) sequence. Therefore, the first step in computational phylogenetic analyses is usually the identification of similar sequences (as an indication of relatedness) and, in particular, the assignment of residues that emerged from a common position in an ancestral sequence through sequence alignment. The initial search of similar sequences usually operates on large databases containing sequences from multiple organisms, and thus, requires fast methods such as `BLAST` [17, 216] or its more modern successors [141]. Homology is inferred from excess similarity which, in turn, is assessed using statistical estimates computed by these tools and appropriate cut-off values [189, 242]. Once a set of (putative) homologs is identified, more accurate alignment methods can be applied. While exact algorithms exist for an optimal global [221] or local [284] alignment of two sequences according to some rule that e.g. aims at minimizing the number of substitutions and insertions/deletions (appearing as gaps in the alignment), exact solutions quickly become infeasible for more than two sequences. Commonly used heuristic tools for the construction of multiple sequence alignments include `Clustal Omega` [281], `MAFFT` [166] and `MUSCLE` [84].

Apart from various metrics for sequence similarity, the distance of two (homologous) sequences can be used to measure how closely they are related. In the simplest case, distance refers to the number of different sites of two aligned sequences which, divided by the length of the alignment, gives the so-called *normalized Hamming distance* or *p distance*. Another commonly applied definition of distance is the expected number of nucleotide substitutions per site [335], which includes mutations that are invisible as a consequence of back-mutations. Clearly, the normalized Hamming distance does not account for the latter, and therefore in general underestimates the number of substitutions that have occurred. As a consequence, more sophisticated, probabilistic methods are required to estimate distances. These are commonly based on the modeling of sequence evolution as a continuous-time Markov chain [88, 335]. Such models assume the so-called *Markovian property*, i.e., the probability of a specific substitution at some site solely depends on the current state of the site (e.g. nucleotide or amino acid) and is independent of the past. The simplest such model for nucleotide substitution is the model by Jukes and Cantor [161] which assumes equal rates of substitution for all nucleotide pairs as well as equal equilibrium frequencies. In contrast, the generalized time-reversible model allows different substitution rates for different nucleotide pairs [311]. In the case of proteins, one usually relies on empirically estimated substitution rate matrices such as in the PAM [66] and the JTT [160] model. Depending on the complexity of the model, estimates for the distance can be obtained using a corresponding formula, e.g. for the Jukes-Cantor-Model, or by means of maximum likelihood estimation (MLE), as usually necessary for the empirically-derived models [335].

The motivation for probabilistic models for sequence evolution is to obtain distances that are proportional to the length of the branch that connects two genes or species in their phylogeny [88]. In other words, one aims to obtain a distance d defined on the pairs in a set X of species or homologous genes that can be represented by a tree T endowed with a map $l\colon E(T) \to \mathbb{R}_{\geq 0}$ of edge lengths. More formally, a distance d is called an *additive (tree) metric* if there is an unrooted tree T with edge lengths l such that, for all $x, y \in X$, $d(x, y) = \sum_{e \in P_{xy}} l(e)$ where P_{xy} is the set of edges on the path from x to y in the tree T [38]. Some of the methods for tree reconstruction discussed in the next section at least implicitly assume *additivity*, i.e., that sequence evolution progresses in a way that can be represented by such an edge-weighted tree.

The *molecular clock* is an even stronger assumption which postulates that mutation rates are (at least approximately) constant in time and across different branches of a phylogeny [343]. As a consequence, distances of genes or species are proportional to the elapsed time since their divergence. The corresponding trees that describe such histories are the so-called *ultrametric* trees [88]. These are rooted trees where the total distance (i.e. the sum of the edge lengths) from the root to any leaf is equal.

3.3.2 *Inference of Trees*

The reconstruction of phylogenetic trees is the key task in the field of phylogenetics. Consequently, a variety of approaches to this problem exist [88, 335]. The starting point (for reconstructions based on molecular data) is usually a multiple sequence alignment (MSA) of nucleic acid or protein sequences which, in the case of gene tree reconstruction, comprises a putative group of homologous genes. For the reconstruction of a species phylogeny, it is common practice to consider multiple loci, usually of well-conserved gene families in which events such as duplications do not or only rarely occur. The sequences of the same species can then be concatenated to obtain one long MSA. Alternatively, so-called *coalescent* methods estimate the species tree and multiple gene trees in a single or two subsequent steps [196, 245]. In the following sections, some classes of methods that rely on an MSA as input are briefly discussed.

Parsimony and Likelihood Methods

The parsimony paradigm is a commonly applied concept in phylogenetics and aims at finding histories with a minimal number of events that explain the observed data. In the reconstruction of trees from MSAs, this amounts to finding a tree that admits a minimal number of substitutions that must have occurred to explain the MSA, a problem known as the *large parsimony problem*. The closely related problem of finding the minimal number of substitutions for a given tree topology is referred to as the *small parsimony problem* and can e.g. be solved efficiently for binary rooted trees using Fitch's algorithm [90] or, given one prefers substitutions with unequal costs, Sankoff's algorithm [256].

In contrast, Graham and Foulds [113] have shown that the large parsimony problem is NP-complete.

The most important aspect causing the "hardness" of this problem is certainly the rapidly growing number of tree topologies for an increasing number of leaves. For example, the number of binary rooted trees with n labeled leaves is given by $(2n-3)!/(2^{n-2}(n-2)!)$ [88]. Therefore, it is usually impractical to enumerate and check all possible trees. Instead, the space of all trees can be explored according to some notion of neighborhood among the trees and the search for a local optimum. In practice, neighborhood is defined in such a way that a tree can be obtained from another tree by one or a few simple operations [88]: *Nearest-neighbor interchanges* delete an inner edge of a binary unrooted tree as well as the four other edges incident to its endpoints and then reconnect the resulting four subtrees in the two ways that differ from the original topology. Another commonly used operation is *subtree pruning and regrafting* which consists of the removal of a whole subtree by deletion of an edge (and the suppression of the degree-two vertex in the remaining tree), followed by the re-attachment of the subtree at a vertex which is newly-created to subdivide some other edge in the remaining tree.

Maximum likelihood methods also rely on a heuristic exploration of the space of possible trees. Instead of minimizing the number of substitutions, they aim at maximizing the likelihood, i.e., the probability to observe the given data (the columns/sites of the MSA). At the same time, the model parameters m and the branch lengths l must be estimated giving the following general equation for the likelihood [241]:

$$L(T) = P(\text{data} \mid T, m, l).$$

In the simplest case, the statistical model consists of a model for sequence evolution discussed in Sec. 3.3.1. However, more complex models that e.g. account for variable substitution rates among different sites are also possible [334]. Maximum likely methods are for example efficiently implemented in `FastTree` [243], `PhyML` [117] and `RAxML` [295].

Bayesian inference methods are also based on probabilistic models for sequence evolution. However, they also include prior distributions for the tree, model parameters, and branch lengths. This allows the computation of posterior probabilities according to Bayes' theorem [88, 241]:

$$P(T, m, l \mid \text{data}) = \frac{P(\text{data} \mid T, m, l) \cdot P(T, m, l)}{P(data)}.$$

Similar to maximum likelihood methods, the tree with a maximum *a posteriori* probability among the resulting ensemble of trees produced by a Bayesian inference method can be chosen as the final result [329]. As a remedy for the infinitely large search space (due to the continuous branch lengths and parameters), Markov Chain Monte Carlo (MCMC) is applied to estimate posterior probabilities. In brief, MCMC comprises sampling algorithms such as the commonly-used Metropolis-Hastings algorithm [126, 213] which, starting from some initial estimate, construct a chain of "states". In phylogeny reconstruction, each state consists of the tree topology, branch lengths, and model parameters [241, 329]. In each step of the chain construction, a new state is proposed according to some perturbation algorithm. The chain either remains in the old state or accepts the new state according to a probability function that depends on the likelihood ratio of the two states, the priors, and a specific correction factor for the sampling approach. After thousands or even millions of steps, the frequency of a tree in the chain is an estimate for its posterior probability. Popular tools that implement Bayesian inference include `MRBAYES` [147] and `BEAST` [302].

Distance-Based Methods

Distance-based methods rely on pre-calculated pairwise sequence distances, which are usually provided in form of a distance matrix. Given an unrooted tree topology T and a distance d on its leaves, the optimal edge lengths l can be calculated efficiently using standard *least squares methods*, i.e., the minimization of the term $\sum_{i=0}^{n} \sum_{j=0}^{n} (d(i,j) - d_T(i,j))^2$ where $d_T(i,j)$ is the sum of the edge lengths on the path connecting the ith and jth leaf [45]. Bandelt and Dress [23] found that, in contrast to the parsimony and probabilistic methods, a unique and optimal (unrooted) tree can be constructed from distance data in polynomial time given the distances are additive.

Clustering methods such as *UPGMA* (short for *Unweighted Pair Group Method with Arithmetic Mean*) [285] and *Neighbor-Joining* [253] are popular and fast distance-based methods.

UPGMA constructs an ultrametric tree on a set X of leaves, and thus, assumes that the distances satisfy a molecular clock. It starts from a set of clusters $\mathcal{C}$ in which each element of X forms its own cluster and with distances $d(C, C')$ that equal the distances of the single elements in $C, C' \in \mathcal{C}$. It then repeatedly merges the two clusters with the smallest distance until only one cluster remains, which corresponds to the root of the resulting tree. In each step the distances of the clusters are updated according to the equation $d(C, C') = 1/(|C| \cdot |C'|) \sum_{x \in C} \sum_{y \in C'} d(x,y)$. There are well-known examples where unequal mutation rates in sister branches lead to a situation in which two leaves with the shortest distance are not connected to the same inner vertex of the tree. In such cases, UPGMA fails to reconstruct the correct tree topology.

Neighbor-Joining starts by initializing the leaves X as the set of vertices (= clusters) under consideration. It accounts for unequal rates across the branches of a tree by constructing, in each step, an auxiliary matrix Q on the vertices according to the formula $Q(i,j) = (n-2)d(i,j) - \sum_{k \neq i} d(i,k) - \sum_{k \neq j} d(j,k)$, and connecting a pair of vertices f and g to a newly-created vertex u that minimize $Q(f,g)$ instead of $d(f,g)$. The branch lengths of the edges fu and gu in the resulting tree can by calculated as $d(f,u) = d(f,g)/2 + (\sum_{k \neq f,g} d(f,k) - \sum_{k \neq f,g} d(g,k))/(2(n-2))$ and, by symmetry, $d(g,u) = d(f,g) - d(f,u)$, respectively. Finally in each step, f and g are removed from the vertices under consideration and replaced by the new vertex u. The distance of u to any other vertex $k \neq f, g$ is calculated as $d(u,k) = (d(f,k) + d(g,k) - d(f,g))/2$. Given the input distance matrix on X was additive, Neighbor-Joining is guaranteed to construct the correct unrooted tree topology and branch lengths [214]. In a naïve implementation, Neighbor-Joining runs in $O(|X|^3)$. However, clever bookkeeping techniques such as e.g. implemented in `QuickTree` [204] and `RapidNJ` [282] can reduce this to a best-case $O(|X|^2)$ running time.

Distance data obtained in real-life analyses often violates additivity (and more often a molecular clock), in particular, if evolution was not tree-like. To this end, generalizations of Neighbor-Joining for phylogenetic networks have been developed, the best-known of which is the `Neighbor-Net` algorithm [36].

Tree Rooting

A common drawback of probabilistic methods based on time-reversible models of sequence evolution and Neighbor-Joining as the probably most popular distance-based method is their inability to infer the position of the root.

There are two main approaches to rooting a tree, recently reviewed in [171]: *outgroup rooting* includes a gene or species in the analysis that is known to have diverged before the last common ancestor of the remaining genes/species, i.e., an *outgroup*. After the construction of an unrooted tree, the leaf corresponding to the outgroup can be discarded while its unique neighbor serves as the root for the remaining tree. The molecular clock predicts the existence of a point in the tree that is equidistant from all leaves. Consequently, *molecular clock rooting* aims at identifying such a point. *Midpoint rooting* [305] can be viewed as one way of tackling this task. It identifies the longest path between two leaves in the tree and places the root in its center. If this center corresponds to a vertex, the latter is declared as the root. Otherwise, the edge containing the center becomes subdivided by a newly-created vertex. Other methods based on a molecular clock include Bayesian approaches as e.g. proposed by Huelsenbeck et al. [148].

Recently, alternative methods for the inference of the root or even whole rooted trees have been proposed that rely on a different kind of data.

Hellmuth et al. [133] start with tree-free estimates of the orthology relation (see also Sec. 3.3.3) of the genes in a set of genomes/species. They demonstrate that cograph editing as a means of noise reduction can be used to obtain a set of event-labeled gene trees. Each individual gene tree may be poorly resolved, depending on the number of duplication events in the respective gene family. However, so-called *informative triples* for the species tree (not to be confused with the informative triples for best match graphs introduced in Chapter 4) can be extracted from these gene trees which, taken together, have been shown to often convey enough information to reconstruct a fully resolved species phylogeny. In particular, this phylogeny is rooted by construction.

Other approaches exploit information conveyed by horizontal gene transfers [65, 306]. In particular, HGT events imply constraints on the (relative) timing in a way that donor and recipient branches in the species tree must have existed at the same time. To this end, Davín et al. [65] start from an unrooted species tree and sample gene families that involve HGT events utilizing undated reconciliation methods. This is followed by the extraction of a maximal subset of transfers that are time-consistent in a dated setting. They demonstrate that the constraints implied by these transfers can be a valuable source of information especially in the reconstruction of microbial phylogenies for which timing information gathered from fossils and stratigraphy is rare.

3.3.3 *Inference of Orthology*

The distinction of orthologs (genes that diverged as a consequence of speciation) from paralogs or xenologs (genes that diverged as a consequence of duplication or have undergone horizontal gene transfer, respectively) is of considerable practical importance in evolutionary biology for several reasons. According to the "ortholog conjecture", orthologous genes in different species are expected to have essentially the same biological and molecular functions, whereas paralogs and xenologs tend to have similar, but distinct functions [222, 309]. This hypothesis is supported by some of the above-mentioned theoretic models for the functional fate of paralogs after duplication such as the neofunctionalization model which predicts duplications to be a major driving force for innovations. In the light of the empirical evidence, however, the ortholog conjecture is controversial. While some studies based on the comparison of the orthology relation with experimentally determined annotations of gene function and gene expression profiles confirm the hypothesis [12, 249], other groups come to the conclusion that the correlation between orthology and preservation of function is much weaker than originally expected [222, 296]. Nevertheless, this assumption is widely made in the computational prediction of gene functions [95, 222, 290, 340]. Reliable predictions, in turn, are the basis for annotating genes in genomes for which experimental data is rare or non-existent, as well as the study of human diseases and gene functions in model organisms such as *Drosophila melanogaster* and *Mus musculus*.

Moreover, the distinction of orthologs and paralogs is crucial in phylogenomics [68], i.e., the reconstruction of species trees from genomic data, since the divergence time of orthologous genes (that are not xenologous in the sense of Def. 3.2) is equal to the divergence time of their respective species.

For these reasons, orthology identification has received increasing attention from both a mathematical and a computational perspective. In particular, a wide array of orthology detection tools has been developed, see e.g. [15, 97, 226, 272, 288] for reviews and applications. In order for researchers to be able to test their methods and compare them with other tools, the *Quest for Orthologs* consortium maintains a web service that provides curated reference proteomes and species trees, as well as a number of standardized methods for orthology benchmarking [14, 16]. One classically distinguishes *tree-based* and *graph-based* methods for orthology detection.

Tree-Based Orthology Inference

Tree-based methods aim at reconstructing event-labeled rooted gene trees. Once the inner vertices of a gene tree are labeled as either a speciation, duplication, or HGT event, it becomes a trivial task to extract the orthology relation (cf. Def. 3.1). An event-labeling, in turn, is naturally implied by a reconciliation between a gene and species tree. In case a reconciliation μ maps onto both vertices and edges of the species tree, the speciation vertices are given by those gene tree vertices that map onto vertices [80, 103, 252]. In some frameworks for reconciliation, on the other hand, such as the DLS-trees in [109] and the DTL-scenarios in [24, 316], the assignment of the event type is directly included in the definition.

Therefore, the actual problem in tree-based orthology inference usually amounts to finding a suitable reconciliation. To this end, it is most common to employ parsimony criteria. In particular, one aims to minimize the number of duplication, loss, and HGT events (possibly weighted by specific costs) necessary to explain the respective reconciliation [106, 115, 237]. Given a pair of rooted gene tree (T, σ) and species tree S, a simple approach to parsimonious reconciliation is the well-known last-common-ancestor (LCA) mapping, which maps every vertex $v \in V(T)$ to the vertex $\operatorname{lca}_S(\sigma(L(T(v))))$, i.e., the last common ancestor in S of the set of species appearing as colors of the leaves in the subtree below v. In the setting of pure duplication-loss scenarios, a vertex is then inferred as a duplication event whenever it has some child that is mapped to the same vertex in S, and as a speciation event otherwise. For binary trees, it was shown that the LCA mapping requires a minimal number of explaining duplication and loss events which can, moreover, be computed in linear time [87, 109, 342]. The permission of HGT events, on the other hand, may admit reconciliations that require an even smaller total number of events. The problem of reconciliation with HGT first arose in the context of host/parasite assemblages [48, 121, 237] where the equivalent problem of mapping a parasite tree T into a host phylogeny S is considered such that the number of events, including host switches (i.e., horizontal transfers), is minimized. The early literature on the topic is reviewed e.g. in [49]. A major difficulty is to enforce time consistency in the presence of multiple horizontal transfer events, which renders the problem of finding optimal reconciliations NP-hard [123, 125, 235, 316].

Commonly used parsimony-based methods and databases for gene and species tree reconciliation include `PhylomeDB` [149] which uses outgroups to root the gene tree, as well as `Ensembl Compara/TreeBeST` [324] and `Orthostrapper` [301] both of which use the parsimony criterion, i.e., the number of duplication-loss events, to root the gene tree. As in gene tree reconstruction, probabilistic models for gene evolution pose an alternative to parsimony approaches. Such a method for orthology detection based on MCMC is implemented in `DLRSOrthology` [20, 320].

Tree-based methods are often considered to be more accurate than graph-based approaches, and thus the methods of choice for smaller datasets [15, 178]. Their applicability in large-scale analyses, however, is limited due to the high computational cost. Other drawbacks include the dependence on the correctness of the input trees, and thus, their sensitivity to phenomena like long attraction which is a common bias in phylogeny reconstruction [303].

Graph-Based Orthology Inference

The results derived in Chapters 4 to 6 of this work primarily fit into second class of methods for orthology inference: Graph-based approaches are based on the idea that, in the absence of horizontal gene transfer, the time at which two species A and B diverged is also the latest possible time at which two genes a and b (from A and B, resp.) may have diverged. Consequently, if a and b are orthologs, then a should be among the closest relatives of b in species A and *vice versa*. The notion of closest relatives in pairs of species can naturally be expressed as a graph on the set of genes, and the symmetry condition results in a certain robustness of these methods. They usually operate in two phases: a graph construction phase followed by a clustering or filtering phase [15].

The graph construction phase builds an initial estimate of the orthology relation based on pairwise sequence comparison. To this end, the closest relatives of a gene in some other species are approximated as those with the highest similarity or smallest distance. In the former case, estimations of symmetrical closest relatives have e.g. been termed symmetrical best hits (symmetrical BeTs) [309], bidirectional best hits (BBH) [236], and reciprocal best hits [32]; and in the latter case, reciprocal smallest distance (RSD) [325]. In order to clearly distinguish between such estimations and the actual closest relatives of a gene in other species, i.e., those with the lowest last common ancestor in the "true" gene tree, we will follow Stadler et al. [293] and use the terms (reciprocal) best hits and (reciprocal) best matches, respectively. We will return to a discussion of this distinction and on how to obtain best matches in Chapter 4.

To obtain similarity scores for pairs of sequences, most tools employ fast heuristic searches such as `BLAST` [17]. Commonly used tools that rely on `BLAST` scores include `InParanoid` [289], `OrthoInspector` [195, 225], `OrthoMCL` [193], and `ProteinOrtho` [189]. Some methods, such as `eggNOG` [150] and `OrthoDB` [179], use scores obtained from the Smith-Waterman algorithm for exact local alignment. The computation of maximum likelihood estimates for evolutionary distances is in general more time-consuming than that of similarity scores. Methods that follow this approach include `OMA` [251, 318] and `Roundup` [69].

The identification of reciprocal best hits is usually followed by a clustering phase or at least some filtering and refinement steps to improve the results. Tatusov et al. [309] proposed a simple method for cluster detection in the estimated graph that consists in the identification and stepwise merging of pairs of triangles that have a common edge. Another approach is *Markov Clustering* as e.g. applied in `OrthoMCL` [193]. This method simulates random walks through the graph from which probabilities that two genes are in the same cluster are estimated. The set of genes is then partitioned according to these probabilities. `ProteinOrtho` attempts to eliminate spurious orthology edges using *Spectral Clustering* [189]. More precisely, the connected components of the graph are partitioned according to the *Fiedler vector*, which is the eigenvector associated with the second smallest eigenvalue of the so-called *Laplacian matrix* of the graph. This process recurses on the resulting components as long as they are below a user-defined threshold for graph connectivity. Finally, some methods determine groups of orthologs by identifying (maximal) cliques in the graph [74, 185]. However, the associated decision problem of deleting at most k edges to obtain a cluster graph, i.e., a graph in which every connected component is a clique, is NP-complete [275], and thus, heuristics are applied.

Some clustering methods, in particular those based on the identification of cliques, assume that orthologs in a pair of species form a one-to-one relationship. However, a duplication event that occurs after a speciation event may lead to the situation that a gene a in species A has two or more orthologs in species B. In this case, the latter are called *co-orthologs* w.r.t. gene a. Hence, the orthology relation as given by Def. 3.1 for a pair of species is a many-to-many relationship in general. In particular, therefore, clustering cannot capture all orthology pairs while, at the same time, excluding paralogous relations. To this end, alternative concepts such as the *clusters of orthologous groups (COGs)* [310] have been introduced. COGs contain all genes that descended from a common ancestor gene in the last common ancestor of the species under consideration. Thus, these groups may also contain paralogs that emerged from a (more recent) duplication event. From a practical perspective, the many-to-many relationship is usually accounted for by the definition of a tolerance threshold up until which sub-optimal (reciprocal) best hits are included as edges in the graph construction phase [150, 189, 195, 251].

Software tools that aim at resolving the orthology relation for pairs of genes instead of only outputting clusters include `InParanoid` [289], `ProteinOrtho` [189] as well as more recent versions of `OMA` [251, 318]. An interesting trade-off between the details of the relationships conveyed by a pairwise resolution and the somewhat more workable groups of orthologs is the inference of *hierarchical groups*, i.e., groups of genes that all descended from a common ancestor w.r.t. some speciation event. By definition, such groups become more and more inclusive starting from the most recent speciation events towards the root of the species tree under consideration. In particular, this naturally implies a bottom-up approach that merges lower-level groups at each inner vertex of a given species tree. This way, corresponding tools such as `Hieranoid` [268] and `OMA GETHOGs` [13, 318] reduce the time complexity by avoiding expensive initial all-vs-all similarity searches between the genomes while also yielding accurate ortholog groups.

Hybrid and Other Methods

Since both tree- and graph-based methods for orthology inference have their specific strengths, mainly an expected higher accuracy and a much lower running time, respectively, it seems natural to combine these advantages in a hybrid approach. For example, the computational pipeline of the `Ensembl Compara` project [324] starts with a `BLAST`-based inference of reciprocal best hits and a clustering phase. Then, a gene tree is built for each cluster and reconciled with a gene tree. Conversely, hierarchical graph-based methods such as `Hieranoid` [268] operate bottom-up along a guide tree and could therefore also be viewed as a hybrid method. The `HyPPO` method (for 'Hybrid Prediction of Paralogs and Orthologs') described in [185] constructs a species tree from an initial estimate of cliques of orthologs which is then used to infer additional orthologs which have undergone accelerated mutation after they emerged from a duplication event and therefore show a smaller similarity to the related genes. In particular, this method exploits the fact that there must be a species tree that allows a reconciliation to explain the ortholog groups.

Similarly, correction methods have been proposed that edit an initial estimate of the orthology relation to a (biologically feasible) cograph using the awareness of an underlying species tree and an event-labeled gene tree that explain the relation [79, 133, 184].

Finally, alternative sources of information (i.e., apart from sequence similarity) can be used for the inference or correction of orthology. In particular, gene order conservation (synteny), which is expected to be indicative for true orthology, has been used for this purpose, see e.g. [182] and tools like `SYNERGY` [326] and the `ProteinOrtho` extension `PoFF` [190].

3.3.4 *Inference of Horizontal Gene Transfer*

A broad spectrum of computational methods have also been developed to identify horizontally transferred genes and/or HGT events, recently reviewed in [246]. Similar to orthology detection, these methods are usually separated into two classes: parametric and phylogenetic methods.

Parametric methods are based on the fact that the genomes of different (groups of) species often differ in certain characteristics ("parameters"). This kind of signature is inherent to the transferred genomic sequence and can therefore mark them as candidate foreign genetic material in the recipient species (whose signature is different). A commonly used characteristic is the nucleotide composition and, in particular the GC content, i.e., the relative abundance of guanine-cytosine base pairs, which varies widely across different species [28, 64, 273]. Similarly, one can use k-mer distributions, sequence autocorrelation, and the so-called codon usage bias which refers to the tendency of certain species to prefer some codons over other synonymous codons, i.e., triplets of nucleotides that encode the same amino acid [25, 83, 188]. Other indicators include structural features of the DNA such as deformability [332] and proximity of sequences to transposases and integrases [120], i.e., genes encoding enzymes that mediate cutting and reinsertion of (often neighboring) DNA sequences. An advantage of parametric methods is the fact that, in principle, only a single genome, the recipient's, needs to be investigated. However, in the reconstruction of gene families, it is also of interest to infer the origin, the donor species, of horizontally transferred genes. Another drawback of these methods is the fact that sequences acquired by HGT will usually adapt to the characteristics of the host genome, a process that is called amelioration [187]. Therefore, parametric methods are usually restricted to the inference of more recent transfers.

Phylogenetic methods for HGT inference are based on the incongruence of species and gene distances or trees. They are further distinguished into explicit and implicit phylogenetic methods [246].

Explicit (or direct) phylogenetic methods start from a given gene tree T and species tree S and compute a reconciliation, i.e., a mapping of the gene tree into the species tree. An edge in T can then be labeled as a horizontal transfer whenever its endpoints are mapped to incomparable vertices/edges of the species tree. Correspondingly, two genes are inferred as xenologous whenever the path in T that connects them contains such an edge. As already mentioned in Sec. 3.3.3, finding an optimal reconciliation involving HGT, i.e., one that minimizes the number of events necessary to explain a scenario, is NP-hard in general. Nevertheless, several practical approaches have become available, see e.g. [51, 200, 316]. Moreover, probabilistic models for sampling reconciliations that involve HGT have been found to produce reliable estimates of transfers [170, 283].

Implicit (or indirect) phylogenetic methods forego the reconstruction of trees and start from sequence similarity or evolutionary distances and use unexpectedly small or large distances between genes as indicators of HGT. The basic idea is that the evolutionary distance between orthologous genes is approximately proportional to the distances between their species. Xenologous gene pairs, as well as duplicate genes, thus appear as outliers [55, 76, 186, 229]. The situation in which two genes diverged earlier than their respective species can be explained by duplication events (and complementary losses if there are no closer related genes), whereas the opposite, i.e., the divergence of the genes being more recent than that of the species, can only be explained by a transfer across incomparable branches of the species tree, and thus, is a clear indicator for HGT. The pairs of genes with this property define the edge set of a graph with the set of genes within a gene family as vertices. One main contribution of this work is a detailed mathematical investigation of the properties of these later-divergence-time (LDT) graphs in Chapters 7 and 8.

A framework to distinguish between additive and replacing HGT was recently described in [176]. Therein, the authors also show that the associated optimal reconciliation-cost problem is NP-complete, and present a simple heuristic that starts with a previously known heuristic for reconciliation involving only additive HGT and then greedily reclassifies transfers as replacing whenever this does not create inconsistencies in the model.

3.4 SIMULATION OF PHYLOGENETIC SCENARIOS

A common problem in phylogenetics is the absence of reliable data sets for benchmarking new methods. Even though standardized methods for benchmarking as in the *quest for orthologs* service provide reliable data sets on the basis of manually curated databases or consensus approaches that combine the results of various methods [14, 16], it cannot be guaranteed that these

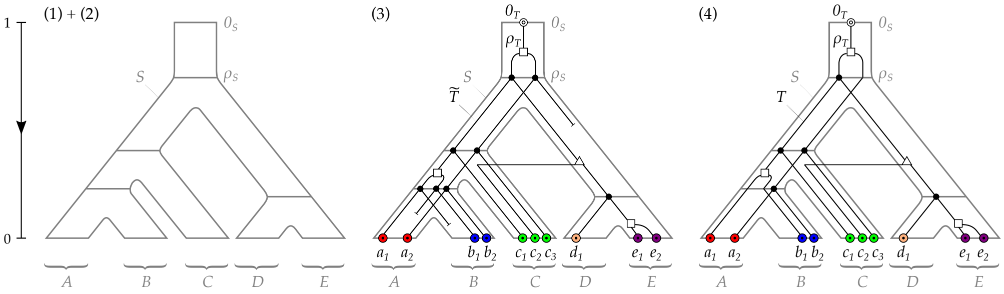


Fig. 5. The general simulation scheme in `AsymmeTree`. See the text and the caption of Fig. 4 for details.

faithfully reflect the true evolutionary histories of the genes in question. Simulated scenarios often provide a remedy to this issue. In particular, they enable researchers to systematically analyze the behavior of their newly developed methods when only selected parameters are changed (such as certain event rates). Therefore, they can give valuable insights into the strengths and weaknesses of a method. However, it is clear that simulations cannot replace benchmarking with biological data for any method that is intended for the application to real-life data.

In order to illustrate the mathematical concepts and methods described in this work and to demonstrate their potential relevance for integration into bioinformatics tools, we will also rely on simulated evolutionary scenarios. To this end, we will use the Python package `AsymmeTree` which was first described in [293] and in some more detail in [258]. `AsymmeTree` can be used to simulate pairs of dated species trees and gene trees involving gene duplications, losses as well as HGT. Commonly used tools for this task include `ALF` [60] and `SimPhy` [207]. However, we will need to be able to manipulate certain parameters and e.g. allow non-binary duplication events. Moreover, `AsymmeTree` provides a number of methods to directly extract statistics and various graphs (BMGs, LDT graphs, etc.) from the simulated scenarios.

General simulation scheme

In brief, `AsymmeTree` generates realistic evolutionary scenarios in four steps (illustrated in Fig. 5): (1) First, a planted species tree S is generated which (2) gets equipped with dating map τ_S. (3) Along S, the event-labeled "true" gene tree $(\widetilde{T}, t, \sigma)$ with dating map τ_T is simulated that also contains duplication, loss, and HGT events according to user-specified rates. In the context of the simulations, the "true" gene tree denotes the gene tree which still contains all branches leading to loss events only. (4) In a final step, the "observable" gene tree is constructed whose leaf set is exactly the set of surviving/extant genes.

The topology of the species tree S is generated using the Innovation Model [168], which models observed phylogenies well. Therein, the branching is driven by innovations that are determined by the gain and loss of so-called features. More precisely, the species are represented by their sets of features. Starting with a set containing a single feature, one selects at random the addition of a novel feature to some set or the deletion of an existing feature from some set. If the resulting feature set is not already among the current set of species/sets, this leads to a branching event. This way, a tree is constructed starting from the root until a user-defined number of leaves is reached. Finally, a planted edge above the root is added to account for the ancestral line, in which gene duplications may already occur. Note that the so-constructed tree does not contain any extinct species branches.

The planted tree S is then equipped with a dating function that assigns timestamps to its vertices. To this end, the tree is traversed in pre-order, i.e., all vertices v are visited before their descendants. If a vertex v is a leaf or the planted root 0_S, then it obtains a time stamp $\tau_S(v) = 0$ or $\tau_S(v) = 1$, respectively. Otherwise, a random leaf x in the subtree $S(v)$ is chosen. Moreover, a real number r is drawn from the uniform distribution on the interval $[0, 2)$. It is then set $\tau_S(v) = \tau_S(\mathsf{par}_S(v)) \cdot r/(p+1)$ where p is the the number of edges on the path from v to the leaf x. By construction, the species tree covers a total time span of one unit in the end. `AsymmeTree` also implements alternative models for the generation of species trees such as the well-known Yule model [339] as well as constant-rate or episodic birth-death processes [122, 169, 294]. All of these methods directly yield a time map τ_S. However, we will exclusively consider the Innovation Model in the simulations throughout this work.

Along the dated species tree (S, τ_S), an event-labeled, dated "true" gene tree $(\widetilde{T}, t, \tau_T, \sigma)$ is simulated using a variant of the constant-rate birth-death process with a given age, see e.g. [122, 169]. To this end, the user must specify rates for the three event types duplication, loss, and HGT which serve as parameters for exponential distributions from which waiting times until the next events are drawn. By setting the respective rate to zero, an event type is disabled completely. The simulation starts with a single gene in the planted root of the species tree and proceeds stepwise in time towards the leaves by drawing waiting times until the next event and branches in which these events take place. At each point in the simulation, the total rate is given by the sum of the three event types over the currently existing branches in the gene tree under construction. Speciations are included as additional branching events that generate copies of all genes present at a speciation vertex in all descendant lineages. In particular, if a waiting time for a duplication, loss, or HGT event is drawn such that the next speciation event would occur earlier, then this waiting time is discarded, the time is updated to the next speciation event, and the latter is executed. If a duplication event is drawn for some branch, then a copy of the gene is placed into the same branch of the species tree. Similarly, for HGT events, a copy is placed into some other branch of the species tree. The recipient branches are chosen at random among the species branches that coexist at the respective point in time. A loss event in some (gene tree) branch results in the creation of a leaf, and the removal of the branch from the set of active branches. An extant gene x corresponds to a branch of $\widetilde{T}$ that reaches present time (i.e., timestamp zero) and thus a leaf s of S, determining $\sigma(x) = s$. All other leaves of $\widetilde{T}$ correspond to losses.

To avoid trivial cases (gene trees with no survivors), `AsymmeTree` implements two different strategies, both of which constrain the loss events. The first one avoids the extinction of the gene family in every branch of the species tree by temporarily setting the loss rate in a branch to zero as long as this gene is the only current survivor in its species branch. In particular, this ensures that every leaf of S has at least one surviving gene. The second method is less restrictive and only forbids the complete extinction of the gene family by setting the loss rate to zero whenever only a single survivor in all species branches is left.

Finally, the "observable" part T of the gene tree $\widetilde{T}$ is obtained by removing all branches that lead to losses only and by suppressing inner vertices with a single child. `AsymmeTree` can also assign rates to edges of T to convert evolutionary time differences into general additive distances. However, this is not relevant here since the rates do not affect evolutionary relatedness or the timing, and thus the simulation analyses considered in this work, which start from the best match graph (Chapters 4 to 6) and the later-divergence-time graph (Chapters 7 and 8), respectively.

Polytomies

Extending the simulations used in [103, 293], we also consider non-binary gene trees. This is important here since, by Lemma 4.15 below, the special motif called hourglass cannot appear in BMGs that are explained by a binary tree. There is an ongoing discussion to what extent polytomies, i.e., vertices with more than two children, in phylogenetic trees are biological reality as opposed to an artifact of insufficient resolution. At the level of species trees, the assumption that cladogenesis occurs by a series of bifurcations [e.g. 73, 201] seems to be prevailing, several authors have argued quite convincingly that there is evidence for a least some *bona fide* multifurcations of species [172, 257, 307]. In the simulation, polytomies in species trees are introduced after the first step by edge contraction with a user-defined probability p.

The reality of polytomies is less clear for gene trees, in particular, the existence of multifurcations apart from those that are "inherited" by the underlying species tree. One reason is the abundance of tandem duplications. Although the majority of tandem arrays comprises only a pair of genes, larger clusters are not at all rare [239]. Although one may argue that mechanistically they likely arise by stepwise duplications, such arrangements are often subject to gene conversion and non-homologous recombination that keeps the sequences nearly identical for some time before they eventually escape from concerted evolution and diverge functionally [124, 194]. As a consequence, duplications in tandem arrays may not be resolvable unless witnesses of different stages of an ongoing duplication process have survived. To model polytomies in the gene tree, we modify step (3) of the simulation procedure by replacing a simple duplication with the generation of $2+k$ offspring genes. The number k of additional copies is drawn from a Poisson distribution with a user-defined parameter $\lambda > 0$. By setting $k = 0$, polytomies at duplication events are disabled.

4

BEST MATCH GRAPHS

Evolutionary closest relatedness among a set of (homologous) genes L is most naturally expressed in terms of last common ancestors of their underlying phylogeny, i.e., the tree (T, σ) that describes their emergence from a single common ancestor through branching events (cf. Sec. 3.1). The *best matches* of a gene x w.r.t. a species $s \neq \sigma(x)$ comprise exactly those genes y in species s such that there is no other gene y' of this species that shares a more recent last common ancestor with x in T. This definition closely ties best matches to the concept of orthology which is also defined in terms of last common ancestors. In particular, graph-based methods for orthology inference approximate best matches by means of sequence similarity or distances as discussed in Sec. 3.3.3. Under the assumption of a strict molecular clock and noise-free data, best hits (i.e., genes with the highest similarity or closest distance) and best matches are equivalent [102]. For real-life data, however, these conditions are often violated, and thus, additional efforts are required to obtain reliable best match estimates, see [293] and Sec. 4.1.4 below for a short summary thereof.

Best matches have several appealing properties. Most importantly, in contrast to best hits, they admit mathematical guarantees for graph-based orthology inference. In particular, all pairs of orthologs are also reciprocal best matches, at least in the absence of HGT [103]. As a consequence, false-negative orthologs are not an issue given one has reliable best match estimates. Moreover, Dessimoz et al. [75] and Geiß et al. [104] discovered that best matches are also sufficient to unambiguously identify certain false-positive ortholog pairs among the reciprocal best matches. We will return to a complete characterization of these pairs in Chapter 6.

Best matches are defined in terms of inequalities w.r.t. the ancestor relation of the underlying gene tree. Therefore, they contain valuable information on the topology of this (usually unknown) tree [102]. If the best matches are estimated with a tree-free approach, e.g. based on best hits, this can be used as an alternative source of information in phylogeny reconstruction compared to the traditional methods discussed in Sec. 3.3.2. Since (reciprocal) best hits are usually inferred using sequence similarity/distance inequalities, they are invariant under monotonic transformations of the similarity or distance measures. This is of great practical relevance in the light of back-mutations and the difficulties arising in the reconstruction of additive distances as they are required by distance-based methods (cf. Secs. 3.3.1–3.3.2). Moreover, since best hits only require considering two genomes at a time, it suffices that the evolutionary rates are approximately the same for all genes in the same species lineage for them to correctly reflect the best match relation.

Gaining a detailed mathematical understanding of best matches is a crucial step on the way to their direct application in orthology or gene tree recon-

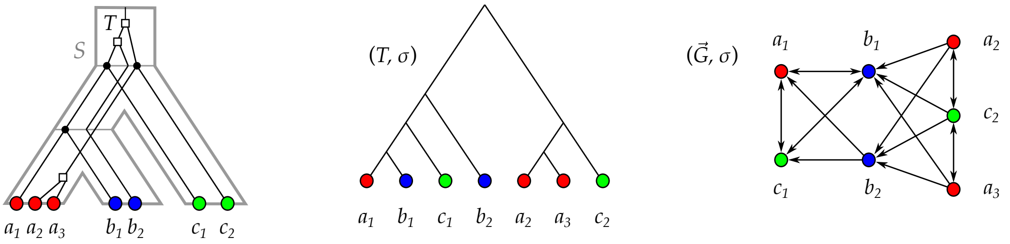


Fig. 6. An evolutionary scenario (left) consisting of a gene tree (T, σ) embedded into a species phylogeny S. For further explanation, see also Fig. 4. The BMG $(\vec{G}, \sigma)$ (right) depends on the topology of the gene tree (middle). A gene y is a best match of x ($x \rightarrow y$ in the BMG) if there is no gene y' of the same color that has a lower last common ancestor with x than y. For instance, b_1 but not b_2 is a best match for a_1 in the blue species.

struction. This chapter establishes a number of novel characterizations of best match graphs (BMGs) and several interesting subclasses. We start by introducing BMGs formally and summarizing the most important previously established results in Sec. 4.1. In Sec. 4.2, we provide two characterizations of general BMGs that are based on certain triple sets and already appeared in [259] and [266]. Sec. 4.3 is entirely dedicated to the subclass of BMGs that can be explained by binary trees. It contains a structural and an algorithmic characterization that have been published previously in [264] and [260], respectively. Finally, Sec. 4.4 is based on [262] and describes a fast algorithm for the recognition of BMGs that contain at most two colors and the construction of their least resolved trees.

4.1 RECENT WORK ON BEST MATCH GRAPHS

4.1.1 *Best Matches, BMGs, and Their Basic Properties*

The combinatorial properties of best matches and their applicability to the correction of orthology estimates have only recently become the topic of in-depth investigations [102, 103, 293], and they were formally introduced by Geiß et al. [102]:

Definition 4.1. *Let (T, σ) be a leaf-colored tree. A leaf $y \in L(T)$ is a* best match *of the leaf $x \in L(T)$ if $\sigma(x) \neq \sigma(y)$ and $\operatorname{lca}_T(x, y) \preceq_T \operatorname{lca}_T(x, y')$ holds for all leaves y' of color $\sigma(y') = \sigma(y)$.*

The colored digraph $\vec{G}(T, \sigma) = ((L, E), \sigma)$ with vertex set $L = L(T)$ and arcs $(x, y) \in E$ if and only if y is a best match of x w.r.t. (T, σ) is known as the (colored) *best match graph* (BMG) of (T, σ) [102]. Correspondingly, we obtain the following class among the properly vertex-colored digraphs:

Definition 4.2. *An arbitrary vertex-colored digraph $(\vec{G}, \sigma)$ is a* best match graph (BMG) *if there exists a leaf-colored tree (T, σ) such that $(\vec{G}, \sigma) = \vec{G}(T, \sigma)$. In this case, we say that (T, σ)* explains *$(\vec{G}, \sigma)$.*

An example for a leaf-colored tree (T, σ) explaining a BMG $(\vec{G}, \sigma)$ together with an evolutionary scenario which may have given rise to the gene phylogeny (T, σ) is given in Fig. 6.

Remark 4.1. *We note that, in contrast to [102], we do not assume that the coloring* $\sigma\colon L \to M$ *in a BMG* $(\vec{G}, \sigma)$ *or a leaf-colored tree* (T, σ) *is a surjective map.*

Since additional "unused" colors by definition have no effect on the best match relation, this assumption was merely made in [102] to simplify the notation, i.e., for writing M instead of $\sigma(L)$ to refer to the full color set. In particular, therefore, the previously established results for BMGs remain valid (except for replacing M by $\sigma(L)$ of course). We omit the surjectivity assumption since it is problematic when subgraphs $(\vec{G}_{|L'}, \sigma_{|L'})$ for a subset $L' \subset L$ are considered that satisfy the conditions of being a BMG except for surjectivity of $\sigma_{|L'}$. We will, however, sometimes speak of r being a color in $(\vec{G}, \sigma)$ or the colors in $(\vec{G}, \sigma)$ to refer to the set $\sigma(L)$, i.e., the colors that actually appear on the vertices, rather than M.

A BMG $(\vec{G} = (L, E), \sigma)$ with $\ell := |\sigma(L)|$ colors will often simply be called *ℓ-BMG*. By construction, no vertex x of a BMG $(\vec{G}, \sigma)$ has a neighbor with the same color, i.e., the coloring σ is proper. Since the last common ancestors of any two vertices of T always exists, and $\operatorname{lca}_T(x, y)$ and $\operatorname{lca}_T(x, z)$ are comparable, there is by definition at least one best match of x for every color $s \in \sigma(L) \setminus \{\sigma(x)\}$:

Observation 4.1. *For every vertex x and every color* $s \in \sigma(L) \setminus \{\sigma(x)\}$ *in a BMG* $(\vec{G} = (L, E), \sigma)$*, there is some vertex* $y \in N(x)$ *with* $\sigma(y) = s$.

Equivalently, the subgraph induced by every pair of colors is sink-free. In particular, therefore, BMGs are sink-free whenever they contain at least two colors. For a characterization of BMGs, it will be useful to formalize this property:

Definition 4.3. *Let* $(\vec{G}, \sigma)$ *be a colored digraph. The coloring σ is* sink-free *if it is proper and, for every vertex x and every color* $s \in \sigma(V(\vec{G})) \setminus \{\sigma(x)\}$*, there is a vertex* $y \in N(x)$ *with* $\sigma(y) = s$*. A digraph with a sink-free coloring is* sf-colored.

In particular, the definition of BMGs together with Obs. 4.1 implies that BMGs are always sf-colored.

Whether two vertices x and y are best matches or not does not depend on the presence or absence of vertices z with $\sigma(z) \notin \{\sigma(x), \sigma(y)\}$. More precisely, we have

Observation 4.2. [102, Obs. 1] *Let* $(\vec{G}, \sigma)$ *be a BMG explained by* (T, σ) *with leaf set L and let* $L' := \bigcup_{s \in M'} L[s]$ *be a subset of vertices with a restricted color set* $M' \subseteq \sigma(L)$*. Then the induced subgraph* $(\vec{G}[L'], \sigma_{|L'})$ *is explained by the restriction* $T_{|L'}$ *of T to the leaf set L', i.e.* $(\vec{G}[L'], \sigma_{|L'}) = \vec{G}(T_{|L'}, \sigma_{|L'})$.

Geiß et al. [102] introduced a set of induced subgraphs of vertex-colored digraphs $(\vec{G}, \sigma)$ associated with certain rooted triples, the *informative triples*, which, in the case of a BMG, contain valuable information on the topologies of all trees that explain $(\vec{G}, \sigma)$. Using Fig. 7, one easily verifies that the definition given in [102] is equivalent to the following simplified version:

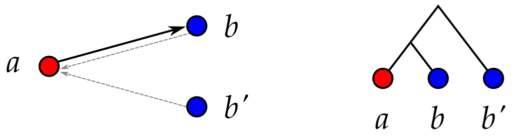


Fig. 7. The informative triple $ab|b' \in \mathcal{R}(\vec{G},\sigma)$ is obtained from the induced subgraph of $(\vec{G},\sigma)$ on the left-hand side. The gray dashed arcs may or may not exist.

Definition 4.4. *Let $(\vec{G},\sigma)$ be a vertex-colored digraph. We say that a triple $ab|b'$ is* informative *for $(\vec{G},\sigma)$ if a, b and b' are pairwise distinct vertices in $\vec{G}$ such that (i) $\sigma(a) \neq \sigma(b) = \sigma(b')$ and (ii) $(a,b) \in E(\vec{G})$ and $(a,b') \notin E(\vec{G})$. The set of informative triples is denoted by $\mathcal{R}(\vec{G},\sigma)$.*

Hence, the set of informative triples of $(\vec{G},\sigma)$ can be expressed as

$$\mathcal{R}(\vec{G},\sigma) = \left\{ab|b' \colon \sigma(a) \neq \sigma(b) = \sigma(b'),\ (a,b) \in E(\vec{G}),\ \text{and}\ (a,b') \notin E(\vec{G})\right\} \tag{3}$$

The term *informative* refers to the following result that all triples in $\mathcal{R}(\vec{G},\sigma)$ are displayed by *every* tree explaining a BMG $(\vec{G},\sigma)$:

Lemma 4.1. *Let $(\vec{G},\sigma)$ be a BMG and $ab|b'$ an informative triple for $(\vec{G},\sigma)$. Then, every tree (T,σ) that explains $(\vec{G},\sigma)$ displays the triple $ab|b'$, i.e.* $\operatorname{lca}_T(a,b) \prec_T \operatorname{lca}_T(a,b') = \operatorname{lca}_T(b,b')$.

Proof. The definition of informative triples implies that $(a,b) \in E(\vec{G})$ and $(a,b') \notin E(\vec{G})$. Using $\sigma(b) = \sigma(b')$ and the definition of best matches, we immediately conclude that $\operatorname{lca}_T(a,b) \prec_T \operatorname{lca}_T(a,b')$. ☐

The structure of a tree (T,σ) can be related to the connectedness of the BMG $\vec{G}(T,\sigma)$ that it explains.

Proposition 4.1. [102, Thm. 1] *Let (T,σ) be a leaf-colored tree and $\vec{G}(T,\sigma)$ its BMG. Then $\vec{G}(T,\sigma)$ is connected if and only if there is a child v of the root ρ_T such that $\sigma(L(T(v))) \neq \sigma(L(T))$. Furthermore, if $\vec{G}(T,\sigma)$ is not connected, then for every connected component $\vec{G}_i$ of $\vec{G}(T,\sigma)$ there is a child v of the root ρ_T such that $V(\vec{G}_i) \subseteq L(T(v))$.*

In other words, a BMG $(\vec{G},\sigma)$ explained by (T,σ) is connected if and only if the root of T has a child v whose subtree $T(v)$ does not contain the full set of colors $\sigma(V(\vec{G})) = \sigma(L(T))$. The following result is closely related.

Lemma 4.2. [102, Prop. 1] *The disjoint union of vertex disjoint BMGs $(\vec{G}_i,\sigma_i)$, $1 \leq i \leq k$, is a BMG if and only if all color sets are the same, i.e., $\sigma_i(V(\vec{G}_i)) = \sigma_j(V(\vec{G}_j))$ for $1 \leq i < j \leq k$.*

In particular, therefore, BMGs can also be understood in terms of their connected components: a digraph $(\vec{G},\sigma)$ is an ℓ-BMG if and only if all of its connected components are ℓ-BMGs. Moreover, an explaining tree for the vertex-disjoint union of BMGs $(\vec{G}_i,\sigma_i)$ with the same color set can easily be constructed from their explaining trees (T_i,σ_i) by joining the latter under a common root.

4.1.2 *Least Resolved Trees*

In general, a BMG $(\vec{G},\sigma)$ can be explained by multiple leaf-colored trees. However, it will be useful to consider those trees among them that are minimal, or *least resolved*, in the following sense:

Definition 4.5. *A leaf-colored tree* (T,σ) *is* least resolved *if there is no non-empty subset* $A \subsetneq E(T)$ *such that* $\vec{G}(T,\sigma) = \vec{G}(T_A,\sigma)$ *where* T_A *is obtained from* T *by contraction of all edges in* A*.*

Thus, a least resolved tree (T,σ) does not display a tree (T',σ) with fewer edges that still explains $\vec{G}(T,\sigma)$. Note that we define the notion of being least resolved here as a property of the tree (T,σ) alone. Of course, every least resolved tree is also *least resolved w.r.t. some BMG*, namely the (uniquely defined) digraph $\vec{G}(T,\sigma)$. Least-resolvedness is closely related to the concept of redundant edges in a tree (T,σ).

Definition 4.6. *An edge* $e \in E(T)$ *is redundant with respect to* $\vec{G}(T,\sigma)$ *if the tree* T_e *obtained by contracting the edge* e *satisfies* $\vec{G}(T_e,\sigma) = \vec{G}(T,\sigma)$*.*

A redundant edge e trivially cannot be an outer edge of a (phylogenetic) tree T, since otherwise, we have $L(T_e) \neq L(T)$, i.e., the two digraphs $\vec{G}(T_e,\sigma_{|L(T_e)})$ and $\vec{G}(T,\sigma)$ do not have the same vertex set, and are therefore not equal. Redundant edges have been characterized in [102] by means of certain equivalence classes, the *thinness classes*, defined on the vertex set of a BMG $(\vec{G},\sigma)$. In Sec. 4.2.2, we will derive a novel characterization that is somewhat simpler and does not need additional notation.

The following result from [102] states that every BMG $(\vec{G},\sigma)$ is explained by a unique least resolved tree, its *LRT*, and relates all other trees that explain $(\vec{G},\sigma)$ via their sets of redundant edges:

Theorem 4.1. [102, Thm. 8] *Every BMG* $(\vec{G},\sigma)$ *is explained by a unique least resolved tree* (T^*,σ)*. The tree* (T^*,σ) *is obtained from an arbitrary tree* (T,σ) *that explains* $(\vec{G},\sigma)$ *by contraction of all redundant edges of* (T,σ)*. In particular, every such tree* (T,σ) *is a refinement of* (T^*,σ)*.*

Therefore, there is a one-to-one correspondence between BMGs and LRTs. Moreover, we immediately obtain

Corollary 4.1. *A tree* (T,σ) *is least resolved if and only if it does not contain redundant edges.*

4.1.3 *Reciprocal Best Match Graphs*

The reciprocal best match graphs (RBMGs) are the symmetric parts of BMGs:

Definition 4.7. *Two leaves* $x,y \in L(T)$ *of a leaf-colored tree* (T,σ) *are* reciprocal best matches *if* y *is a best match of* x *and vice versa.*

An undirected graph (G,σ) *is a* reciprocal best match graph (RBMG) *if it is the symmetric part of some BMG* $\vec{G}(T,\sigma)$*. In this case, we say that* (T,σ) explains (G,σ)*.*

Conceptually, reciprocal best matches correspond to the reciprocal best hits used in orthology detection. In contrast to BMGs, RBMGs are much more difficult to handle and are not associated with unique trees [104].

RBMGs with at most three colors can be recognized in polynomial time as shown in [104, Lemma 30 and 31] and [137, Cor. 1]. The complexity of deciding whether a graph (G, σ) with more than three colors is an RBMG is an open problem. Nevertheless, it was shown in [137] that the decision version of the deletion and editing problem to obtain a valid RBMG is NP-hard for $\ell \geq 2$ colors. In particular, therefore, it is NP-complete for two and three colors.

RBMGs that are cographs, or co-RBMGS for short, can be recognized in polynomial time since they admit a characterization as *hierarchically colored cographs*, a certain class of vertex-colored cographs whose coloring satisfies a particular recursive construction rule [104, Thm. 11]. Intriguingly, 2-colored RBMGs are always cographs [137, Cor. 1] and, for an RBMG with more colors, it suffices to check whether all of its 3-colored induced subgraphs are cographs to decide whether the full graph is a cograph as well [104, Thm. 8]. Otherwise, at least one of the 3-colored induced subgraphs contains a 3-colored P_4 as an induced subgraph. These P_4s fall into three distinct classes: the good, the bad, and the ugly quartets [104]. These are determined by the order of the colors in the P_4 and the additional one-directional arcs between the four vertices in the corresponding BMG. Good quartets have already been shown to indicate false-positive orthology assignments [75, 103]. We will show in Chapter 6 that this is the case for all three types of quartets.

4.1.4 *Inference of Best Matches*

The inference of best matches as those genes in some other species with the smallest distance becomes exact when the distance is an ultrametric on the set of extant genes, i.e., when sequences evolve under a strict molecular clock [102, 293, 343]. However, many models for the functional fate of the gene copies predict unequal substitution rates after they emerged from a duplication event [93, 231]. Indeed, experimental evidence also suggests that asymmetric divergence of paralogs is a common phenomenon across the tree of life [21, 42, 70, 72, 127].

As a consequence, best match estimates based on best hits are burdened with systematic errors, i.e., false predictions arising from varying evolutionary rates across the tree rather than simple noise. Of course, the latter is also unavoidable due to sequencing and alignment errors as well as limited sequence lengths. Even if the measured distances are free of noise and perfectly additive, both false-negative and false-positive best match assignments are possible as the examples in Fig. 8 show.

Stadler et al. [293] suggest several strategies to deal with unequal mutation rates in the inference of best matches.

The simplest approach is the definition of a relative threshold parameter ϵ. More precisely, sub-optimal hits in a species Y are inferred as best matches of x if their distance is not worse than a factor $1+\epsilon$ than that of the closest

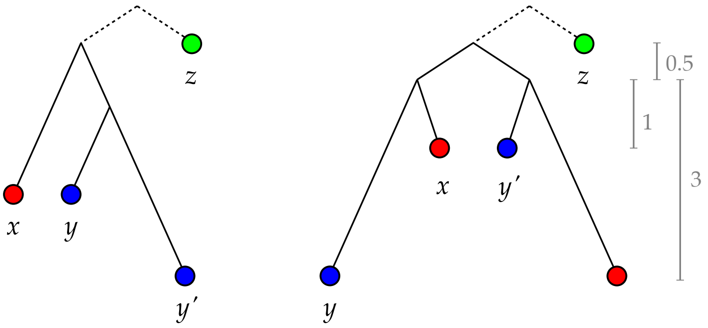


Fig. 8. Examples for the incongruence of best hits and best matches. The lengths of the edges indicate evolutionary distances. On the l.h.s., both y and y' are best matches of x but the distance to y is much smaller. Hence, (x, y') is a false-negative best match. On the r.h.s., the smallest distance of 3 results in the false-positive best match assignments (x, y') and (y', x). For approximately additive distances, the inclusion of an outgroup gene z in the analysis often reveals the correct best match relation in such situations. Adapted from [258].

gene of x in species Y. A very similar approach defining a relative threshold for sequence similarity is e.g. implemented in `ProteinOrtho` [189] in order to account for co-orthologs. Depending on the choice of ϵ and the extent of asymmetric divergence in the data set under consideration, the inclusion of sub-optimal best hits can correctly deal with situations such as on the l.h.s. of Fig. 8. However, the false-positive assignments (x, y') and (y', x) on the r.h.s. cannot be corrected with this approach.

The "quartet" method provides a solution to this issue. It requires the knowledge of one or more reliable outgroup genes z for every gene x and its best match candidates in some species $Y(\neq \sigma(x))$. By considering x, z and two candidate genes from Y, a so-called *quartet*, i.e., one of the four possible unrooted trees (including the star tree) with four labeled leaves, can be inferred based on their pairwise distances. This only requires that the distance is just close enough to being additive such that the correct quartet is inferred. In the example on the l.h.s. of Fig. 8, the quartet on $\{x, y, y', z\}$ is the unrooted tree with an inner edge uv such that x and z are adjacent to u, whereas y and y' are adjacent to v. Since z is a known outgroup gene, this implies a unique triple on the remaining three genes ($yy'|x$ in the l.h.s. example). By considering all candidate pairs in species Y and with the help of an auxiliary digraph, the best matches of x can be identified in this way. A problem that remains is the choice of suitable outgroup genes z. To this end, the authors of [293] suggest using homologous genes from outgroup species as a heuristic. Hence, only additional (partial) information on the topology of the species tree is required. They show that this method yields suitable outgroup genes in the majority of cases with the exception of some (presumably rare) scenarios that include multiple duplication events in the ancestral line of the species under consideration.

4.2.1 *Useful Properties*

We start by collecting some useful properties of BMGs and RBMGs that will be needed for later reference.

Observation 4.3. *Let (T,σ) be a tree explaining the BMG $(\vec{G},\sigma)$, and $v \in V(T)$ a vertex such that $\sigma(L(T(v))) = \sigma(L(T))$. Then $(a,b) \in E(\vec{G})$ and $a \in L(T(v))$ implies $b \in L(T(v))$.*

The following two closely related results on bidirectional arcs in BMGs (and thus edges in the corresponding RBMGs) will be needed regularly in this and the following chapters.

Lemma 4.3. [104, Lemma 10] *Let (T,σ) be a leaf-colored tree on L and let $v \in V(T)$. Then, for any two distinct colors $r,s \in \sigma(L(T(v)))$, there are arcs $(x,y),(y,x)$ in $\vec{G}(T,\sigma)$ with $x \in L[r] \cap L(T(v))$ and $y \in L[s] \cap L(T(v))$.*

Lemma 4.4. *Let $(\vec{G},\sigma)$ be a BMG explained by a tree (T,σ). Moreover, let $x,y \in L(T)$ with $\sigma(x) \neq \sigma(y)$ and $v_x, v_y \in \mathsf{child}_T(\mathsf{lca}_T(x,y))$ with $x \preceq_T v_x$ and $y \preceq_T v_y$. Then, $\sigma(x) \notin \sigma(L(T(v_y)))$ and $\sigma(y) \notin \sigma(L(T(v_x)))$ if and only if $(x,y),(y,x)$ are arcs in $\vec{G}$.*

Proof. By the definition of best matches, it holds that $(x,y),(y,x)$ are arcs in $\vec{G}$ if and only if $\mathsf{lca}_T(x,y) \preceq_T \mathsf{lca}_T(x,y')$ for all $y' \in L(T)$ of color $\sigma(y)$ and $\mathsf{lca}_T(x,y) \preceq_T \mathsf{lca}_T(x',y)$ for all $x' \in L(T)$ of color $\sigma(x)$. Clearly, $\mathsf{lca}_T(x,y) \preceq_T \mathsf{lca}_T(x,y')$ for all such y' if and only if $\sigma(y) \notin \sigma(L(T(v_x)))$, and $\mathsf{lca}_T(x,y) \preceq_T \mathsf{lca}_T(x',y)$ for all such x' if and only if $\sigma(x) \notin \sigma(L(T(v_y)))$. □

The next result is concerned with pairs of "overlapping" informative triples and the positions of the involved leaves that they imply for every explaining tree.

Lemma 4.5. *Let $ab|b'$ and $cb'|b$ be informative triples for a BMG $(\vec{G},\sigma)$. Then every tree (T,σ) that explains $(\vec{G},\sigma)$ contains two distinct children $v_1, v_2 \in \mathsf{child}_T(\mathsf{lca}_T(a,c))$ such that $a,b \prec_T v_1$ and $b',c \prec_T v_2$.*

Proof. Let (T,σ) be an arbitrary tree that explains $(\vec{G},\sigma)$. By Lemma 4.1, T displays the informative triples $ab|b'$ and $cb'|b$. Thus we have $\mathsf{lca}_T(a,b) \prec_T \mathsf{lca}_T(a,b') = \mathsf{lca}_T(b,b')$ and $\mathsf{lca}_T(c,b') \prec_T \mathsf{lca}_T(c,b) = \mathsf{lca}_T(b,b')$. In particular, $\mathsf{lca}_T(a,b') = \mathsf{lca}_T(b,b') = \mathsf{lca}_T(c,b) =: u$. Therefore, $a \preceq_T v_1$ and $b' \preceq_T v_2$ for distinct $v_1, v_2 \in \mathsf{child}_T(u)$. Since $\mathsf{lca}_T(a,b) \prec_T u$, we have $a,b \prec_T v_1$ and thus v_1 is an inner vertex. Likewise, $\mathsf{lca}_T(b',c) \prec_T u$ implies $b',c \prec_T v_2$. □

4.2.2 *Least Resolved Trees*

Edges in leaf-colored trees (T,σ) that are redundant w.r.t. their BMGs $\vec{G}(T,\sigma)$ have already been characterized in [102, Lemma 15, Thm. 8] in terms of certain equivalence classes using a rather complicated notation. Here we give a simpler characterization:

Lemma 4.6. *Let* $(\vec{G},\sigma)$ *be a BMG explained by a tree* (T,σ)*. The edge* $e = uv$ *with* $v \prec_T u$ *in* (T,σ) *is redundant w.r.t.* $(\vec{G},\sigma)$ *if and only if (i)* e *is an inner edge of* T *and (ii) there is no arc* $(a,b) \in E(\vec{G})$ *such that* $\operatorname{lca}_T(a,b) = v$ *and* $\sigma(b) \in \sigma(L(T(u)) \setminus L(T(v)))$*.*

Proof. Let w_e be the vertex in T_e resulting from the contraction $e = uv$ with $v \prec_T u$ in T. By assumption we have $(\vec{G},\sigma) = \vec{G}(T,\sigma)$.

First, assume that e is redundant and thus, $\vec{G}(T_e,\sigma) = \vec{G}(T,\sigma)$. Then e must be an inner edge, since otherwise $L(T) \neq L(T_e)$ and, therefore, (T_e,σ) does not explain $(\vec{G},\sigma)$. Now assume, for contradiction, that there is an arc $(a,b) \in E(\vec{G})$ such that $\operatorname{lca}_T(a,b) = v$ and $\sigma(b) \in \sigma(L(T(u)) \setminus L(T(v)))$. Then there is a leaf $b' \in L(T(u)) \setminus L(T(v))$ with $\sigma(b') = \sigma(b)$ and $\operatorname{lca}_T(a,b) = v \prec_T u = \operatorname{lca}_T(a,b')$. Thus, $(a,b') \notin E(\vec{G})$. After contraction of e, we have $\operatorname{lca}_T(a,b) = \operatorname{lca}_T(a,b') = w_e$. Hence, by definition of best matches, (a,b) is an arc in $\vec{G}(T_e,\sigma)$ if and only if (a,b') is an arc in $\vec{G}(T_e,\sigma)$; a contradiction to the assumption that (T_e,σ) explains $(\vec{G},\sigma)$.

Conversely, assume that $e = uv$ with $v \prec_T u$ is an inner edge in T and that there is no arc $(a,b) \in E(\vec{G})$ such that $\operatorname{lca}_T(a,b) = v$ and $\sigma(b) \in \sigma(L(T(u)) \setminus L(T(v)))$. In order to show that an edge e is redundant, we need to verify that $\vec{G}(T,\sigma) = \vec{G}(T_e,\sigma)$. To this end, consider an arbitrary leaf $c \in L(T)$. Then we have either Case (1) $c \in L(T) \setminus L(T(v))$, or Case (2) $c \in L(T(v))$.

In Case (1) it is easy to verify that $\operatorname{lca}_T(c,d) = \operatorname{lca}_{T_e}(c,d)$ for every $d \in L(T)$. In particular, therefore, $(c,d) \in E(\vec{G}(T,\sigma))$ if and only if $(c,d) \in E(\vec{G}(T_e,\sigma))$.

In Case (2), i.e. $c \in L(T(v))$, consider another, arbitrary, leaf $d \in L(T)$. Note, if $\sigma(c) = \sigma(d)$, then c and d never form a best match. Thus, we assume $\sigma(c) \neq \sigma(d)$. Now, we consider three mutually exclusive Subcases (a) $\operatorname{lca}_T(c,d) \preceq_T v$, (b) $\operatorname{lca}_T(c,d) = u$ and (c) $\operatorname{lca}_T(c,d) \succ_T u$.

Case (a). Since no edge below v is contracted, we have for every d' with $\sigma(d') = \sigma(d)$, $\operatorname{lca}_T(c,d') \prec_T \operatorname{lca}_T(c,d) \preceq_T v$ if and only if $\operatorname{lca}_{T_e}(c,d') \prec_{T_e} \operatorname{lca}_{T_e}(c,d) \preceq_{T_e} w_e$. In particular, therefore, $(c,d) \in E(\vec{G}(T,\sigma))$ if and only if $(c,d) \in E(\vec{G}(T_e,\sigma))$.

Case (b). $\operatorname{lca}_T(c,d) = u$ and $c \prec_T v$ implies that $d \in L(T(u) \setminus L(T(v))$ and thus, $\sigma(d) \in \sigma(L(T(u)) \setminus L(T(v)))$. If $(c,d) \in E(\vec{G}(T,\sigma))$, then $\sigma(d) \notin \sigma(L(T(v)))$ must hold. Therefore, (c,d) is still an arc after contraction of e. For the case $(c,d) \notin E(\vec{G}(T,\sigma))$, assume for contradiction $(c,d) \in E(\vec{G}(T_e,\sigma))$. Then $(c,d) \notin E(\vec{G}(T,\sigma))$ implies that there must be a vertex d' with $\sigma(d') = \sigma(d)$ and $\operatorname{lca}_T(c,d') \preceq_T v \prec_T u = \operatorname{lca}_T(c,d)$. In particular, $d' \in L(T(v))$ can be chosen such that $\operatorname{lca}_T(c,d')$ is farthest away from v and thus, $(c,d') \in E(\vec{G}(T,\sigma))$. Now, $\operatorname{lca}_T(c,d') \preceq_T v$ and $(c,d) \in E(\vec{G}(T_e,\sigma))$ imply that $\operatorname{lca}_{T_e}(c,d') = w_e = \operatorname{lca}_{T_e}(c,d)$, which is only possible if $\operatorname{lca}_T(c,d') = v$. In summary, we found an arc $(c,d') \in E(\vec{G}(T,\sigma))$ with $\operatorname{lca}_T(c,d') = v$ and $\sigma(d') \in \sigma(L(T(u)) \setminus L(T(v)))$; a contradiction to our assumption. Hence, in Case (b) we have $(c,d) \in E(\vec{G}(T,\sigma))$ if and only if $(c,d) \in E(\vec{G}(T_e,\sigma))$.

Case (c). Since $\operatorname{lca}_T(c,d) \succ_T u$, it is again easy to see that, for every d' with $\sigma(d') = \sigma(d)$, $\operatorname{lca}_T(c,d') \prec_T \operatorname{lca}_T(c,d)$ if and only if $\operatorname{lca}_{T_e}(c,d') \prec_{T_e} \operatorname{lca}_{T_e}(c,d)$ and thus, $(c,d) \in E(\vec{G}(T,\sigma))$ if and only if $(c,d) \in E(\vec{G}(T_e,\sigma))$.

In summary, we have $(c,d) \in E(\vec{G}(T,\sigma))$ if and only if $(c,d) \in E(\vec{G}(T_e,\sigma))$ for all $c,d \in L(T)$. Thus, e is redundant. ☐

Recall that an inner edge $e = uv$ with $v \prec_T u$ of a rooted tree T is distinguished by a triple $ab|c \in r(T)$ if $\operatorname{lca}_T(a,b) = v$ and $\operatorname{lca}_T(a,b,c) = u$. Lemma 4.6 immediately implies the following generalization of Lemma 13 in [102]:

Corollary 4.2. *Let* $(\vec{G},\sigma)$ *be a BMG explained by a tree* (T,σ)*. An inner edge* e *of* (T,σ) *is non-redundant w.r.t.* $(\vec{G},\sigma)$ *if and only if it is distinguished by an*

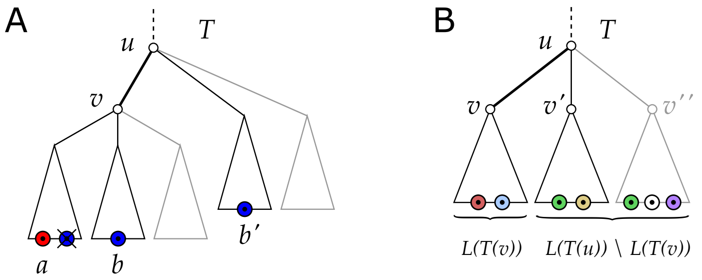


Fig. 9. Visualization of (A) a *non*-redundant edge uv for Lemma 4.6 and (B) a redundant edge uv as in Cor. 4.3. The gray subtrees may or may not exist. In (A), the crossed out leaf indicates that the blue color must not be present in this subtree and thus (a,b) is a best match. In (B), $\sigma(L(T(v)))$ must not have elements in common with $\sigma(L(T(u))\setminus L(T(v)))$. See text for further details.

informative triple $ab|b'$ for $(\vec{G},\sigma)$. In particular, if (T,σ) is least resolved, then each of its inner edges is distinguished by an informative triple.

As another consequence of Lemma 4.6, we obtain

Corollary 4.3. *Let (T,σ) be a leaf-colored tree explaining $(\vec{G},\sigma)$ and uv an inner edge inner of T with $v \prec_T u$. If $\sigma(L(T(v))) \cap \sigma(L(T(v'))) = \emptyset$ for every $v' \in \mathsf{child}_T(u)\setminus\{v\}$, then uv is redundant in T (w.r.t. $(\vec{G},\sigma)$).*

Proof. If there is an arc $e = (a,b) \in E(\vec{G})$ with $\mathrm{lca}_T(a,b) = v$ we have $\sigma(b) \notin L(T(u))\setminus L(T(v)) = \cup_{v'\in\mathsf{child}_T(u)\setminus\{v\}} L(T(v'))$ because $\sigma(L(T(v)))\cap\sigma(L(T(v'))) = \emptyset$ for every $v' \in \mathsf{child}_T(u)\setminus\{v\}$. By Lemma 4.6, the inner edge uv is redundant. $\square$

Both Lemma 4.6 and Cor. 4.3 are illustrated in Fig. 9: In (A), uv is a non-redundant inner edge since (a,b) is a best match such that a and b have v as their last common ancestor and the color of b is present in another subtree below vertex u. Contraction of the edge uv would result in a tree T_{uv} in which $\mathrm{lca}_{T_{uv}}(a,b) = \mathrm{lca}_{T_{uv}}(a,b')$, and thus, introduce the additional best match (a,b'). Clearly, this cannot occur whenever the other subtrees of u do not share any colors with the subtree $T(v)$, a situation that is shown in (B), i.e., the edge uv is redundant w.r.t. the BMG $\vec{G}(T,\sigma)$.

Finally, we show that redundant edges can be contracted in an arbitrary order, similar to [102, Lemma 6 & Cor. 2]. To this end, we first prove a more general statement.

Lemma 4.7. *If T_A is obtained from T by contracting all edges in a subset A of inner edges in T, then $\vec{G}(T,\sigma) \subseteq \vec{G}(T_A,\sigma)$.*

Proof. First note that $L(T_A) = L(T)$ since A only contains inner edges. Let (x,y) be an arc in $\vec{G}(T,\sigma)$. This implies that there is no y' with $\sigma(y') = \sigma(y)$ such that $\mathrm{lca}_T(x,y') \prec_T \mathrm{lca}_T(x,y)$. It is easy to verify that the latter is still true after contraction of an arbitrary edge e, i.e. there is no y' with $\sigma(y') = \sigma(y)$ such that $\mathrm{lca}_{T_e}(x,y') \prec_{T_e} \mathrm{lca}_{T_e}(x,y)$. Hence, (x,y) is an arc in $\vec{G}(T_e,\sigma)$. Now consider the subsets $A_1 \subset A_2 \subset \dots \subset A_{|A|} = A$ where each $|A_i| = i$, $1 \le i \le |A|$. The argument above implies $\vec{G}(T,\sigma) \subseteq \vec{G}(T_{A_1},\sigma) \subseteq \dots \subseteq \vec{G}(T_A,\sigma)$, which completes the proof. $\square$

Lemma 4.8. *Let A and B be disjoint sets of redundant edges in (T,σ) w.r.t. $(\vec{G},\sigma)$ and denote by T_A the tree obtained by contraction of all edges in A in arbitrary order. Then B is a set of redundant edges in T_A w.r.t. $\vec{G}(T_A,\sigma) = \vec{G}(T,\sigma)$.*

Proof. By Lemma 4.7, contraction of *any* inner edge $e = uv \in E(T)$ never leads to a loss of arcs in the BMG $(\vec{G},\sigma) = \vec{G}(T,\sigma)$. Furthermore, the redundant edges in T w.r.t. $(\vec{G},\sigma)$ are completely characterized by Lemma 4.6. Thm. 8 in [102] states that by contraction of all redundant edges (in an arbitrary order), one obtains the unique least resolved tree (T^*,σ) of $(\vec{G},\sigma)$. As argued above, no arc of $\vec{G}(T,\sigma)$ can be lost in the stepwise contraction of redundant edges. Together with $\vec{G}(T,\sigma) = \vec{G}(T^*,\sigma) = (\vec{G},\sigma)$ this implies $\vec{G}(T_A,\sigma) = (\vec{G},\sigma)$. Since by assumption $A \cap B = \emptyset$ and $A \cup B$ is a set of redundant edges w.r.t. $(\vec{G},\sigma)$, we have $(T_A)_B = T_{A\cup B}$ and $\vec{G}(T_A,\sigma) = (\vec{G},\sigma) = \vec{G}(T_{A\cup B},\sigma) = \vec{G}((T_A)_B,\sigma)$. Hence, B is a set of redundant edges in T_A w.r.t. $\vec{G}(T_A,\sigma)$. ☐

Somewhat surprisingly, the property of being least resolved is preserved under contraction of inner edges of a tree (T,σ).

Proposition 4.2. *Suppose (T,σ) is least resolved and let A be a set of inner edges of T, and denote by T_A the tree obtained from a tree T by contracting all edges in A. Then (T_A,σ) is again least resolved.*

Proof. Assume that (T,σ) is least resolved, i.e., it does not contain any redundant edges, and set $(\vec{G},\sigma) := \vec{G}(T,\sigma)$. Lemma 4.6 implies that an inner edge $e = uv$ with $v \prec_T u$ in (T,σ) is non-redundant if and only if there is an arc $(a,b) \in E(\vec{G})$ such that $\operatorname{lca}_T(a,b) = v$ and $\sigma(b) \in \sigma(L(T(u)) \setminus L(T(v)))$. The statement trivially holds if (T,σ) has at most one inner edge. Hence, we assume that (T,σ) has at least two distinct inner edges $e = uv$ and e'. We show that every non-redundant edge e in T remains non-redundant in $T_{e'}$. Thus, let e be a non-redundant edge in T. Hence, there is an arc $(a,b) \in E(\vec{G})$ such that $\operatorname{lca}_T(a,b) = v$ and $\sigma(b) \in \sigma(L(T(u)) \setminus L(T(v)))$. Now consider the tree $T_{e'}$ obtained from T by contraction of the inner edge $e' \neq e$. Clearly, we also have $\operatorname{lca}_{T_{e'}}(a,b) = v$ and $\sigma(b) \in \sigma(L(T_{e'}(u)) \setminus L(T_{e'}(v)))$. Lemma 4.7 implies $\vec{G}(T,\sigma) \subseteq \vec{G}(T_{e'},\sigma)$, and thus, $(a,b) \in E(\vec{G}(T_{e'},\sigma))$. Making again use of Lemma 4.6, we conclude that e is non-redundant in $(T_{e'},\sigma)$.

Since both e and e' were chosen arbitrarily, we observe that the contraction of a single inner edge does not produce new redundant edges. We can therefore apply this argument for each step in the consecutive contraction of all edges in A (in an arbitrary order) to conclude that (T_A,σ) does not contain redundant edges. Therefore, Cor. 4.1 implies that (T_A,σ) is least resolved. ☐

Corollary 4.4. *If (T,σ) is least resolved and A is a non-empty set of inner edges of T, then $\vec{G}(T,\sigma) \subsetneq \vec{G}(T_A,\sigma)$.*

Proof. By Lemma 4.7, we have $\vec{G}(T,\sigma) \subseteq \vec{G}(T_A,\sigma)$. By Prop. 4.2, (T_A,σ) is least resolved. Since the LRT of a BMG is unique (cf. Thm. 4.1), we have $\vec{G}(T,\sigma) \neq \vec{G}(T_A,\sigma)$. ☐

As another immediate consequence of Prop. 4.2 and uniqueness of the LRT of a BMG (Thm. 4.1), we obtain

Corollary 4.5. *If e and e' are two distinct inner edges of a least resolved tree (T,σ), then $\vec{G}(T_e,\sigma) \neq \vec{G}(T_{e'},\sigma)$.*

The following technical result relates subtrees and induced subgraphs of BMGs.

Lemma 4.9. *Let (T,σ) be a tree explaining a BMG $(\vec{G},\sigma)$. Then $\vec{G}(T(u),\sigma_{|L(T(u))}) = (\vec{G}[L(T(u))],\sigma_{|L(T(u))})$ holds for every $u \in V(T)$. Moreover, if (T,σ) is least resolved for $(\vec{G},\sigma)$, then the subtree $T(u)$ is least resolved for $\vec{G}(T(u),\sigma_{|L(T(u))})$.*

Proof. Let $(\vec{G}_1,\sigma') := \vec{G}\left(T(u),\sigma_{|L(T(u))}\right)$ and $(\vec{G}_2,\sigma') := (\vec{G}[L(T(u))],\sigma_{|L(T(u))})$. By definition, we have $V(\vec{G}_1) = V(\vec{G}_2) = L(T(u))$. First assume that $(x,y) \in E(\vec{G}_1)$ for some $x,y \in L(T(u))$. Hence, it holds that $\operatorname{lca}_{T(u)}(x,y) \preceq_{T(u)} \operatorname{lca}_{T(u)}(x,y')$ for all y' with $\sigma(y) = \sigma(y')$ in $T(u)$ and thus, since $T(u)$ is a subtree of T, we have $\operatorname{lca}_T(x,y) \preceq_T \operatorname{lca}_T(x,y')$ for all y' with $\sigma(y) = \sigma(y')$ in T. Therefore, $(x,y) \in E(\vec{G})$. Since $x,y \in L(T(u))$ and $\vec{G}_2$ is the subgraph of $\vec{G}$ induced by $L(T(u))$, we have $(x,y) \in E(\vec{G}_2)$ and thus $E(\vec{G}_1) \subseteq E(\vec{G}_2)$. Now assume $(x,y) \in E(\vec{G}_2)$ for some $x,y \in L(T(u))$. Hence, $(x,y) \in E(\vec{G})$. Consequently, there is no leaf y' in T with $\sigma(y') = \sigma(y) \neq \sigma(x)$ such that $\operatorname{lca}_T(x,y') \prec_T \operatorname{lca}_T(x,y) \preceq_T u$. This clearly also holds for the subtree $T(u)$. Therefore, we have $(x,y) \in E(\vec{G}_1)$ and thus $E(\vec{G}_2) \subseteq E(\vec{G}_1)$.

The second part of the statement is trivially satisfied if $T(u)$ does not contain any inner edges, which is exactly the case if either $u \in L(T)$ or $u \in V^0(T)$ with $\operatorname{child}_T(u) \subseteq L(T)$. Thus, let $u \in V^0(T)$ and $\operatorname{child}_T(u) \cap V^0(T) \neq \emptyset$. Since (T,σ) is least resolved, it does not contain redundant edges. Let vw be an inner edge of $T(u)$ with $w \prec_T v \preceq_T u$, and note that vw must also be an inner edge in T. By Lemma 4.6 and since vw is not redundant in T, there is an arc $(a,b) \in E(\vec{G})$ such that $\operatorname{lca}_T(a,b) = w$ and $\sigma(b) \in \sigma(L(T(v)) \setminus L(T(w)))$. Since $u \succeq_T v$, the arguments above imply that (a,b) is also an arc in $\vec{G}(T(u),\sigma_{|L(T(u))})$ and $\operatorname{lca}_{T(u)}(a,b) = v$. Hence, in particular, we have $\sigma(b) \in \sigma(L(T(v)) \setminus L(T(w)))$. We can now apply Lemma 4.6 to conclude that vw is not redundant in $T(u)$. Since vw was chosen arbitrarily, we conclude that $T(u)$ does not contain any redundant edge and thus, it must be least resolved for $\vec{G}(T(u),\sigma_{|L(T(u))})$ for all $u \in V(T)$. □

Interestingly, all (proper) subtrees $T(u)$ of a least resolved tree explain a connected BMG:

Lemma 4.10. *Let $(\vec{G},\sigma)$ be a BMG and (T,σ) its least resolved tree. Then the BMG $\vec{G}(T(v),\sigma_{|L(T(v))})$ is connected for every $v \in V(T)$ with $v \prec_T \rho_T$.*

Proof. Set $L_v := L(T(v))$. By Lemma 4.9, $\vec{G}(T(v),\sigma_{|L_v})$ is a BMG. First observe that the BMG $\vec{G}(T(v),\sigma_{|L_v})$ is trivially connected if v is a leaf. Now let $v \prec_T \rho_T$ be an arbitrary inner vertex of T. Thus, there exists a vertex $u \succ_T v$ such that uv is an inner edge. Since (T,σ) is least resolved, it does not contain any redundant edges. Hence, by contraposition of Lemma 4.6, there is an arc $(a,b) \in E(\vec{G})$ such that $\operatorname{lca}_T(a,b) = v$ and $\sigma(b) \in \sigma(L(T(u)) \setminus L_v)$. Since $a,b \in L_v$, Lemma 4.9 implies that (a,b) is also an arc in $\vec{G}(T(v),\sigma_{|L_v})$. Moreover, $\operatorname{lca}_{T(v)}(a,b) = v$ clearly also holds in the subtree rooted at v. Now consider the child $w \in \operatorname{child}_{T(v)}(v)$ such that $a \preceq_{T(v)} w$. There cannot be a leaf $b' \in L(T(w))$ with $\sigma(b') = \sigma(b)$ since otherwise $\operatorname{lca}_{T(v)}(a,b') \preceq_{T(v)} w \prec_{T(v)} v$ would contradict that (a,b) is an arc in $\vec{G}(T(v),\sigma_{|L_v})$. Thus $\sigma(b) \notin \sigma(L(T(w)))$. Since $\sigma(b) \in \sigma(L_v)$, we thus conclude $\sigma(L(T(w))) \neq \sigma(L_v)$. The latter together with Prop. 4.1 implies that $\vec{G}(T(v),\sigma_{|L_v})$ is connected. □

The converse of Lemma 4.10, however, is not true, i.e., a tree (T,σ) for which $\vec{G}(T(v),\sigma_{|L_v})$ is connected for every $v \in V(T)$ with $v \prec_T \rho_T$ is not necessarily least resolved. To see this, consider the caterpillar tree (T,σ) given by $(x'',(x',(x,y)))$ with $\sigma(x) = \sigma(x') = \sigma(x'') \neq \sigma(y)$ and $u = \operatorname{lca}_T(x,x')$. It is an easy task to verify that the BMG of each subtree of T is connected. However, the edge $\rho_T u$ is redundant.

4.2.3 *Characterization via Informative Triples*

In this section, we derive a new characterization of best match graphs based on informative triples and the `BUILD` algorithm. In particular, we address a necessary correction of a similar result, Theorem 9, in [102]. Therein, it was claimed that a connected vertex-colored digraph $(\vec{G} = (L, E), \sigma)$ is a BMG if all of its 2-colored induced subgraphs are BMGs and the union

$$R := \bigcup_{s,t \in \sigma(L)} r(T^*_{st}) \tag{4}$$

of all triples obtained from the least resolved tree T^*_{st} of its 2-colored induced subgraphs $(\vec{G}_{st}, \sigma_{st})$ is consistent. More precisely, the original result reads as follows:

> *"A connected colored digraph* $(\vec{G} = (L, E), \sigma)$ *is a BMG if and only if (i) all induced subgraphs* $(\vec{G}_{st}, \sigma_{st})$ *on two colors are 2-BMGs, and (ii) the union R of all triples obtained from their least resolved trees* (T_{st}, σ_{st}) *forms a consistent set. In particular,* $(\mathrm{Aho}(R, L), \sigma)$ *is the unique least resolved tree that explains* $(\vec{G}, \sigma)$*."*

We shall see in Prop. 4.5 below that $\vec{G}(\mathrm{Aho}(R, L), \sigma)$ is always a subgraph of $(\vec{G} = (L, E), \sigma)$ whenever R is consistent. The example in Fig. 10 shows, however, that $\vec{G}(\mathrm{Aho}(R, L), \sigma) \neq (\vec{G}, \sigma)$ is possible because $\mathrm{Aho}(R, L)$ can contain triples that are not present in any of the 2-colored trees (T_{st}, σ_{st}).

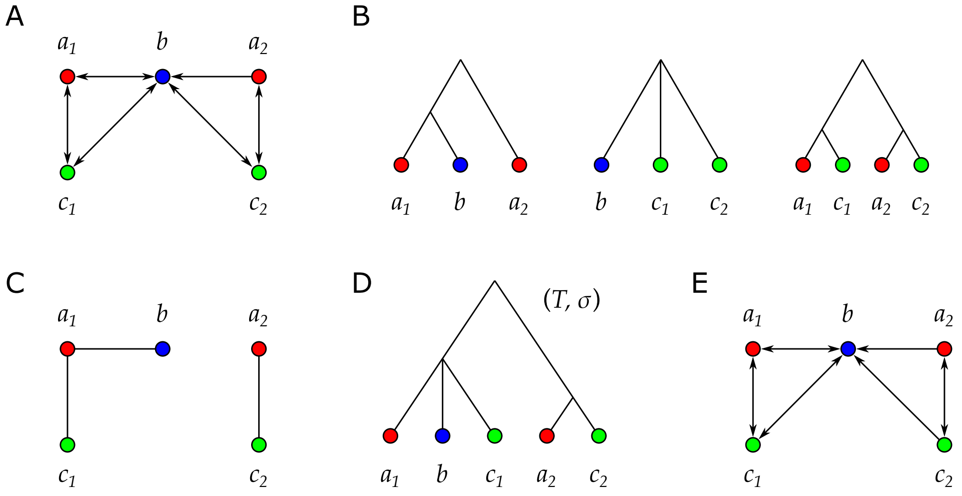


Fig. 10. Counterexample for Thm. 9 in [102]. (A) A vertex-colored digraph with vertex set L that is not a (3-)BMG. (B) The least resolved trees for the three 2-colored induced subgraphs. The union of their triples is $R := \{a_1b|a_2,\ a_1c_1|a_2,\ a_1c_1|c_2,\ a_2c_2|a_1,\ a_2c_2|c_1\}$. (C) The Aho graph $[R, L]$. In particular, R forms a consistent set. (D) The tree $T := \mathrm{Aho}(R, L)$. (E) The 3-BMG $\vec{G}(T, \sigma)$. The arc (b, c_2) that was present in $(\vec{G}, \sigma)$ is missing in $\vec{G}(T, \sigma)$.

As a consequence, the characterization of BMGs requires the equality $(\vec{G}, \sigma) = \vec{G}(\mathrm{Aho}(R, L), \sigma)$ as an additional condition. However, instead of proving an amended version of Thm. 9 in [102], we will state and prove a slightly stronger and more convenient result, Thm. 4.2 below.

To this end, we will first provide several technical results that make use of of (non-)redundant tree edges and, in particular, of least resolved trees. We note that the proofs of Lemma 4.1 and 4.6 above only require the definition of best match graphs, and are thus independent of the results proved in [102].

We can relate the subtrees $T(u)$ to the construction of the Aho graph in the following way:

Lemma 4.11. *Let* (T,σ) *be least resolved for a BMG* $(\vec{G},\sigma)$ *with informative triple set* $\mathcal{R} := \mathcal{R}(\vec{G},\sigma)$. *Then,* $L(T(v))$ *is a connected component in the Aho graph* $[\mathcal{R}_{|L(T(u))}, L(T(u))]$ *for every inner vertex* u *and each of its children* $v \in \mathsf{child}_T(u)$.

Proof. We proceed by induction on $L := V(\vec{G})$. The statement trivially holds for $|L| = 1$. Hence, suppose that $|L| > 1$ and assume that the statement is true for every BMG with less than $|L|$ vertices.

Let u be an inner vertex of T and $v \in \mathsf{child}_T(u)$. We first show that $L(T(v))$ is connected in $[\mathcal{R}_{|L(T(u))}, L(T(u))]$, and then argue that there are no edges between $L(T(v))$ and $L(T(u)) \setminus L(T(v))$, i.e., that $L(T(v))$ forms a connected component.

If uv is an outer edge, i.e., v is a leaf, then $L(T(v))$ is trivially connected. Now suppose that uv is an inner edge of T. By Lemma 4.9, $(\vec{G}[L(T(v))], \sigma_{|L(T(v))})$ is explained by the least resolved tree $(T(v), \sigma_{|L(T(v))})$. By the induction hypothesis, $L(T(w))$ forms a connected component in $[\mathcal{R}_{|L(T(v))}, L(T(v))]$ for all children $w \in \mathsf{child}_T(v)$. Together with $\mathcal{R}_{|L(T(v))} \subseteq \mathcal{R}_{|L(T(u))}$, this implies that the elements in $L(T(w))$ are also connected in $[\mathcal{R}_{|L(T(u))}, L(T(u))]$ for all $w \in \mathsf{child}_T(v)$. Since uv is an inner edge of the least resolved tree (T,σ), we can apply Cor. 4.2 to conclude that there is an informative triple $ab|b'$ in $(\vec{G},\sigma)$ that distinguishes uv, i.e. $\mathrm{lca}_T(a,b) = v$ and $b' \in L(T(u)) \setminus L(T(v))$ with color $\sigma(b') = \sigma(b)$. Hence, $ab|b'$ is also contained in $[\mathcal{R}_{|L(T(u))}, L(T(u))]$. In particular, there are children $w, w' \in \mathsf{child}_T(v)$ such that $a \preceq_T w$ and $b \preceq_T w'$, and the edge ab connects $L(T(w))$ and $L(T(w'))$ in $[\mathcal{R}_{|L(T(u))}, L(T(u))]$.

Now suppose that there is an additional child $w'' \in \mathsf{child}_T(v) \setminus \{w, w'\}$. We distinguish two cases. Either there is a leaf $b'' \preceq_T w''$ with $\sigma(b'') = \sigma(b)$ or no such leaf exists. If there is such a leaf b'', then (a, b'') forms an arc in $(\vec{G},\sigma)$ and $ab''|b'$ is an informative triple making $L(T(w))$ and $L(T(w''))$ connected in $[\mathcal{R}_{|L(T(u))}, L(T(u))]$. Otherwise, take an arbitrary leaf $c \preceq_T w''$. Since $\sigma(b) \notin \sigma(L(T(w'')))$, we have $\sigma(c) \neq \sigma(b)$ and thus, there is an arc (c, b) in $(\vec{G},\sigma)$. Since $\mathrm{lca}_T(c, b') = u \succ_T v = \mathrm{lca}_T(c, b)$, the arc (c, b') is not contained in $(\vec{G},\sigma)$. Hence, $cb|b'$ is an informative triple making $L(T(w'))$ and $L(T(w''))$ connected in $[\mathcal{R}_{|L(T(u))}, L(T(u))]$.

Therefore, the subgraph in $[\mathcal{R}_{|L(T(u))}, L(T(u))]$ induced by $L(T(v))$ must be connected.

It remains to show that $L(T(v))$ is a connected component in $[\mathcal{R}_{|L(T(u))}, L(T(u))]$ and thus, that there are no edges ab in $[\mathcal{R}_{|L(T(u))}, L(T(u))]$ with $a \in L(T(v))$ and $b \in L(T(u)) \setminus L(T(v))$. Assume, for contradiction, that there exists such an edge ab. Hence, this edge must be supported by an informative triple w.l.o.g. $ab|b'$ with $\sigma(a) \neq \sigma(b) = \sigma(b')$ and $b' \in L(T(u))$. Lemma 4.1 implies that $ab|b'$ must be displayed by T. However, $\mathrm{lca}_T(a,b) = u = \mathrm{lca}_T(a,b,b')$ implies that such a triple cannot exist. Thus, $L(T(v))$ is a connected component in $[\mathcal{R}_{|L(T(u))}, L(T(u))]$. ☐

The least resolved tree of a BMG therefore coincides with the Aho tree of its informative triples. In more detail, we have

Proposition 4.3. *If* $(\vec{G} = (L,E), \sigma)$ *is a BMG, then* $(\mathrm{Aho}(\mathcal{R}(\vec{G},\sigma), L), \sigma)$ *is the unique least resolved tree for* $(\vec{G},\sigma)$.

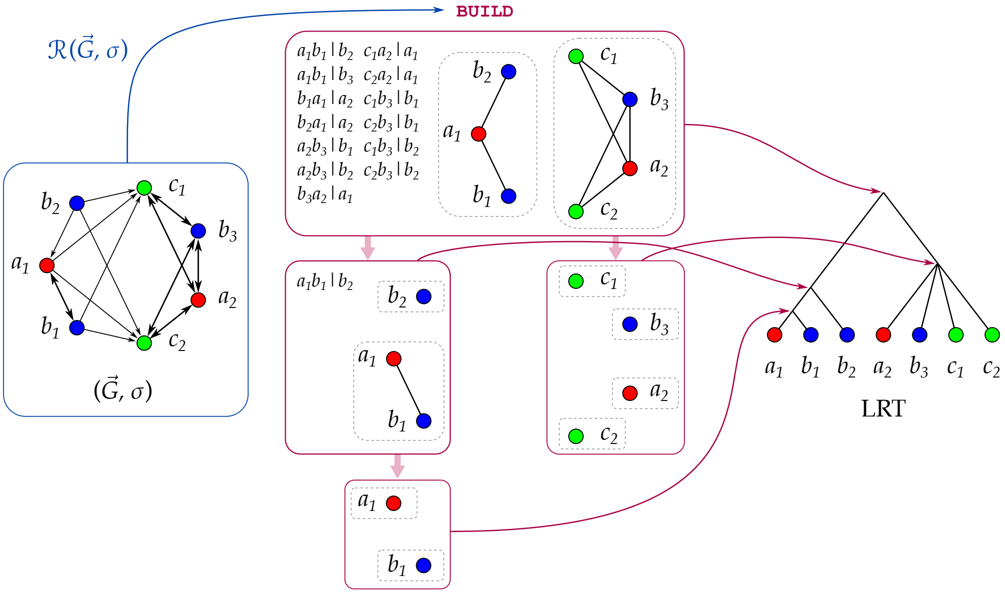

BUILD
$\mathcal{R}(\vec{G},\sigma)$
$a_1b_1|b_2$ $c_1a_2|a_1$
$a_1b_1|b_3$ $c_2a_2|a_1$
$b_1a_1|a_2$ $c_1b_3|b_1$
$b_2a_1|a_2$ $c_2b_3|b_1$
$a_2b_3|b_1$ $c_1b_3|b_2$
$a_2b_3|b_2$ $c_2b_3|b_2$
$b_3a_2|a_1$
$a_1b_1|b_2$
$(\vec{G},\sigma)$
a_1 b_1 b_2 a_2 b_3 c_1 c_2
LRT

Fig. 11. Construction of the LRT for a BMG $(\vec{G},\sigma)$ using the `BUILD` algorithm. Each recursion step (pink boxes) corresponds to a vertex of the resulting tree (trivial steps on single vertices are omitted in the drawing). The algorithm recurses on the connected components (gray dashed boxes) of the Aho graphs and the corresponding subsets of triples.

Proof. Since $(\vec{G},\sigma)$ is a BMG, Lemma 4.1 implies that there is a tree displaying all triples in $\mathcal{R}(\vec{G},\sigma)$. In particular, therefore, $\mathrm{Aho}(\mathcal{R}(\vec{G},\sigma),L)$ exists. Moreover, there must be a least resolved tree (T^*,σ) for $(\vec{G},\sigma)$. To see this, consider an arbitrary tree (T,σ) that explains $(\vec{G},\sigma)$, and repeatedly identify and contract a redundant edge until no redundant edges remain. By definition, the resulting tree still explains $(\vec{G},\sigma)$ and is least resolved. By Lemma 4.11 and by construction of $(\mathrm{Aho}(\mathcal{R}(\vec{G},\sigma),L),\sigma)$, any least resolved tree (T^*,σ) for $(\vec{G},\sigma)$ coincides with the latter. The uniqueness of $\mathrm{Aho}(\mathcal{R}(\vec{G},\sigma),L)$ therefore implies that the least resolved tree is also unique. ☐

We now have all the pieces in place to formulate and prove the main result of this section:

Theorem 4.2. *A colored digraph* $(\vec{G}=(L,E),\sigma)$ *is a BMG if and only if* $\vec{G}(\mathrm{Aho}(\mathcal{R}(\vec{G},\sigma),L),\sigma)=(\vec{G},\sigma)$. *Moreover,* $(\mathrm{Aho}(\mathcal{R}(\vec{G},\sigma),L),\sigma)$ *is the unique least resolved tree explaining a BMG* $(\vec{G},\sigma)$.

Proof. If $(\vec{G}=(L,E),\sigma)$ is a BMG, then Prop. 4.3 implies that $(T:=\mathrm{Aho}(\mathcal{R}(\vec{G},\sigma),L),\sigma)$ is its unique least resolved tree, and thus $\vec{G}(T,\sigma)=(\vec{G},\sigma)$. Conversely, $(\vec{G}(T,\sigma),\sigma)$ is a BMG. ☐

By Thm. 4.2, the set of informative triples $\mathcal{R}(\vec{G},\sigma)$ of a BMG $(\vec{G},\sigma)$ is consistent. In particular, it can be used to construct its LRT by means of the `BUILD` algorithm, see Fig. 11 for an example.

None of the intermediate results used to prove Thm. 9 in [102] is used in our proof of Thm. 4.2. It is worth noting, therefore, that Thm. 4.2 immediately implies Thms. 5, 6, and 7, as well as the existence of a unique least resolved tree in Thms. 2 and 8 of [102]. In particular, Thm. 4.2 allows us to obtain the least resolved tree of a BMG without the need to explicitly construct the least resolved trees of all its 2-colored induced subgraphs, and is not restricted to connected digraphs.

For completeness, we provide two additional results concerning the triple set R as defined in Eq. (4). To this end, we will need that, given a digraph $(\vec{G}, \sigma)$ for which R exists, Lemma 4.1 implies that

$$\mathcal{R}(\vec{G}, \sigma) \subseteq R. \tag{5}$$

Proposition 4.4. *If* $(\vec{G} = (L, E), \sigma)$ *is a BMG, then* $\operatorname{Aho}(\mathcal{R}(\vec{G}, \sigma), L) = \operatorname{Aho}(R, L)$.

Proof. For brevity set $\mathcal{R} := \mathcal{R}(\vec{G}, \sigma)$. From Eq. (5), i.e., $\mathcal{R} \subseteq R$, we immediately have $\mathcal{R}_{|L(T(u))} \subseteq R_{|L(T(u))}$ for every inner vertex u of T. Moreover, by Thm. 4.2, (T, σ) with $T := \operatorname{Aho}(\mathcal{R}, L)$ is the least resolved tree that explains $(\vec{G}, \sigma)$.

Hence, we can apply the same arguments as in the proof of Lemma 4.11 to conclude that $L(T(v))$ forms a connected component in the Aho graph $[R_{|L(T(u))}, L(T(u))]$ for every inner vertex u and each of its children $v \in \mathsf{child}_T(u)$. More precisely, note that connectedness of any such $L(T(v))$ is guaranteed by the informative triples. Now assume, for contradiction, that there is an edge ab in $[R_{|L(T(u))}, L(T(u))]$ with $a \in L(T(v))$ and $b \in L(T(u)) \setminus L(T(v))$ connecting $L(T(v))$ and $L(T(v'))$ for some child $v' \in \mathsf{child}_T(u) \setminus \{v\}$. In this case, there is a triple $ab|c \in R_{|L(T(u))}$ and thus, $a, b, c \in L(T(u))$ and $\operatorname{lca}_T(a, b, c) = u$. By definition of R and Observation 4 in [102], $ab|c$ must be displayed by T. However, $a, b, c \in L(T(u))$ and $\operatorname{lca}_T(a, b) = u = \operatorname{lca}_T(a, b, c)$ imply that $ab|c$ is not displayed by T; a contradiction. Therefore, $(T, \sigma) = (\operatorname{Aho}(R, L), \sigma)$, which completes the proof. ☐

Moreover, existence and consistency of R and $\mathcal{R}(\vec{G}, \sigma)$ guarantees that the BMGs of their Aho graphs are subgraphs of any vertex-colored digraph $(\vec{G}, \sigma)$:

Proposition 4.5. *Let* $(\vec{G} = (L, E), \sigma)$ *be a properly colored digraph with all 2-colored induced subgraphs being 2-BMGs. Then the following two statements hold:*

(i) If $\mathcal{R}(\vec{G}, \sigma)$ *is consistent, then* $\vec{G}(\operatorname{Aho}(\mathcal{R}(\vec{G}, \sigma), L), \sigma) \subseteq (\vec{G}, \sigma)$.

(ii) If R *is consistent, then* $\vec{G}(\operatorname{Aho}(R, L), \sigma) \subseteq (\vec{G}, \sigma)$.

Proof. We set $(\vec{G}', \sigma) := \vec{G}(\operatorname{Aho}(\mathcal{R}(\vec{G}, \sigma), L), \sigma)$. Since $\operatorname{Aho}(\mathcal{R}(\vec{G}, \sigma), L)$ is defined on L, we have $V(\vec{G}') = L = V(\vec{G})$. Now assume, for contradiction, that there is an arc $(a, b) \in E(\vec{G}')$ such that $(a, b) \notin E(\vec{G})$. By assumption, the induced subgraph $(\vec{G}_{st}, \sigma_{st})$ of (G, σ), where $s = \sigma(a)$ and $t = \sigma(b)$, is a 2-BMG and thus sink-free. Therefore, there must be a vertex b' of color $\sigma(b)$ with $(a, b') \in E(\vec{G})$. Hence, $ab'|b$ is informative for $(\vec{G}, \sigma)$ and contained in $\mathcal{R}(\vec{G}, \sigma)$. In particular, $ab'|b$ must be displayed by $\operatorname{Aho}(\mathcal{R}(\vec{G}, \sigma), L)$; contradicting that (a, b) is an arc in $(\vec{G}', \sigma)$. Hence, statement (i) is true.

Statement (ii) can be shown using Eq. (5), i.e., $\mathcal{R}(\vec{G}, \sigma) \subseteq R$, and arguments similar to the previous paragraph. ☐

Finally, we discuss the consequences of the corrected characterization of BMGs for the algorithmic aspects outlined in Sec. 5 of [102]. Regarding the recognition of BMGs, we have noted above that the consistency of the triple set R and the fact that all 2-colored induced subgraphs are 2-BMGs are not sufficient. Algorithm 1 of [102] therefore also needs to be corrected. By Thm. 4.2, it suffices to construct the tree $T := \operatorname{Aho}(\mathcal{R}(\vec{G}, \sigma), L)$ and to check whether

$\vec{G}(T,\sigma) = (\vec{G},\sigma)$. On the other hand, it is no longer necessary to require connectedness of the input digraph. We therefore obtain a considerably simpler procedure, see Alg. 1.

Algorithm 1: Unique least resolved tree of a BMG.

```
Input: Properly colored digraph $(\vec{G} = (L,E),\sigma)$.
Output: Least resolved tree $(T,\sigma)$ if $(\vec{G},\sigma)$ is a BMG, false otherwise.
$T \leftarrow \mathrm{Aho}(\mathcal{R}(\vec{G},\sigma), L)$
if $\vec{G}(T,\sigma) = (\vec{G},\sigma)$ then
    return $(T,\sigma)$
else
    return false
```

The same arguments as in [102] show that $T = \mathrm{Aho}(\mathcal{R}(\vec{G},\sigma), L)$ can be constructed in $O(|E||L|\log^2(|E||L|)) = O(|E||L|\log^2|L|)$ time using the algorithm by Deng and Fernández-Baca [71]. The construction of $\vec{G}(T,\sigma)$ can then be achieved in $O(|L|^2)$ time e.g. using Algorithm 1 in the Supplement of [103]. The equality $\vec{G}(T,\sigma) = (\vec{G},\sigma)$ can be checked in $O(|L|^2)$ operations. The total effort therefore remains dominated by the construction of the least resolved tree T. Therefore, we obtain

Corollary 4.6. *Alg. 1 determines in polynomial time whether a vertex-colored digraph $(\vec{G},\sigma)$ is a BMG. In the affirmative case, a (least resolved) tree that explains $(\vec{G},\sigma)$ can be constructed in polynomial time.*

We note that Algorithm 3 in [102] is essentially the simplified Algorithm 1 above with its input restricted to 2-colored connected digraphs. Its correctness therefore follows immediately from Thm. 4.2.

4.2.4 *Characterization via Informative and Forbidden Triples*

In this section, we provide a second novel characterization of BMGs utilizing forbidden triples as an additional source of information as compared to that provided by the informative triples. This characterization will in particular be useful for understanding BMGs that can be explained by binary trees (Sec. 4.3) and for the derivation of an integer linear programming formulation for BMG editing (Sec. 5.4). The following set of rooted triples obtained from a vertex-colored digraph will be essential.

Definition 4.8. *Let $(\vec{G},\sigma)$ be a vertex-colored digraph. We say that a triple $xy|y'$ is* forbidden *for a digraph $(\vec{G},\sigma)$ if x, y and y' are pairwise distinct vertices in $\vec{G}$ such that (i) $\sigma(x) \neq \sigma(y) = \sigma(y')$ and (ii) $(x,y) \in E(\vec{G})$ and $(x,y') \in E(\vec{G})$. The set of forbidden triples of $(\vec{G},\sigma)$ is denoted by $\mathcal{F}(\vec{G},\sigma)$.*

For short, the set of forbidden triples of $(\vec{G},\sigma)$ can be expressed as

$$\mathcal{F}(\vec{G},\sigma) = \left\{ ab|b' \colon \sigma(a) \neq \sigma(b) = \sigma(b'),\ b \neq b',\ \text{ and } (a,b),(a,b') \in E(\vec{G}) \right\} \tag{6}$$

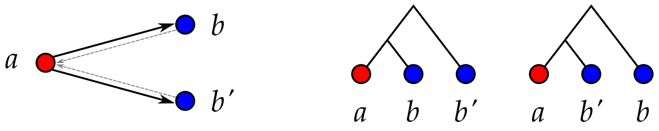


Fig. 12. The forbidden triples $ab|b', ab'|b \in \mathcal{F}(\vec{G},\sigma)$ are obtained from the induced subgraph of $(\vec{G},\sigma)$ on the left-hand side. The gray dashed arcs may or may not exist.

and is illustrated in Fig. 12. As a direct consequence of their definition, forbidden triples always come in pairs:

Observation 4.4. *Let $(\vec{G},\sigma)$ be a vertex-colored digraph. Then $ab|b' \in \mathcal{F}(\vec{G},\sigma)$ with $\sigma(b)=\sigma(b')$ if and only if $ab'|b \in \mathcal{F}(\vec{G},\sigma)$.*

We emphasize that a BMG $(\vec{G},\sigma)$ can contain arcs $(a,b),(a,b')$ with $\sigma(a) \neq \sigma(b) = \sigma(b')$. In this case, however, a tree that explains $(\vec{G},\sigma)$ cannot display the triple $ab|b'$, as shown in

Lemma 4.12. *Let $(\vec{G},\sigma)$ be a BMG explained by (T,σ). Then, none of the triples in $\mathcal{F}(\vec{G},\sigma)$ is displayed by (T,σ).*

Proof. Assume, for contradiction, that (T,σ) displays $ab|b' \in \mathcal{F}(\vec{G},\sigma)$. Hence, $\operatorname{lca}_T(a,b) \prec_T \operatorname{lca}_T(a,b')$ and thus, $(a,b') \notin E(\vec{G})$ contradicting the definition of $ab|b'$ as a forbidden triple of $(\vec{G},\sigma)$. □

Lemma 4.13. *Let $(\vec{G},\sigma)$ be an sf-colored digraph with vertex set L. Then for every tree (T,σ) on L displaying all triples in $\mathcal{R}(\vec{G},\sigma)$, it holds $E(\vec{G}(T,\sigma)) \subseteq E(\vec{G})$.*

Proof. Let (T,σ) be a tree displaying all triples in $\mathcal{R}(\vec{G},\sigma)$ and set $E' := E(\vec{G}(T,\sigma))$ and $E := E(\vec{G})$. First note that $(\vec{G},\sigma)$ and $\vec{G}(T,\sigma)$ have the same vertex set L. Suppose that $(a,b) \in E'$ but $(a,b) \notin E$. Since $(\vec{G},\sigma)$ is sf-colored, vertex a must have at least one out-neighbor b' (distinct from b) of color $\sigma(b)$ in $(\vec{G},\sigma)$, i.e. $(a,b') \in E$. Hence, $(a,b') \in E$ and $(a,b) \notin E$ imply that $ab'|b$ is an informative triple of $(\vec{G},\sigma)$ and thus displayed by T. Therefore $\operatorname{lca}_T(a,b') \prec_T \operatorname{lca}_T(a,b)$ which, together with $\sigma(b)=\sigma(b')$, contradicts $(a,b) \in E'$. Therefore, $E' \subseteq E$. □

Proposition 4.6. *Let $(\vec{G},\sigma)$ be a sf-colored digraph with vertex set L. A leaf-colored tree (T,σ) on L explains $(\vec{G},\sigma)$ if and only if (T,σ) agrees with $(\mathcal{R}(\vec{G},\sigma),\mathcal{F}(\vec{G},\sigma))$. In this case, $(\vec{G},\sigma)$ is a BMG.*

Proof. First suppose that (T,σ) explains $(\vec{G},\sigma)$, in which case $(\vec{G},\sigma)$ is a BMG. The only-if-direction now immediately follows from Lemmas 4.1 and 4.12.

Now suppose that there is a tree (T,σ) on L that displays all triples in $\mathcal{R}(\vec{G},\sigma)$ and none of the triples in $\mathcal{F}(\vec{G},\sigma)$. Hence, we can apply Lemma 4.13 to conclude that $E' := E(\vec{G}(T,\sigma)) \subseteq E(\vec{G}) =: E$. Note that $(\vec{G},\sigma)$ and $\vec{G}(T,\sigma)$ have the same vertex set L. We show that $E' = E$. Assume, for contradiction, that $E' \subset E$, and thus, that there is an $(a,b) \in E \setminus E'$. By Obs. 4.1 and since $\vec{G}(T,\sigma)$ is a BMG, vertex a must have at least one out-neighbor b' of color $\sigma(b)$. Hence, there is an arc $(a,b') \in E'$. Thus, $ab'|b$ is an informative triple of $\vec{G}(T,\sigma)$ and must therefore be displayed by T. Moreover, $(a,b') \in E'$ and $E' \subset E$ imply $(a,b') \in E$. Hence, $(a,b),(a,b') \in E$ implies that $ab'|b$ is a forbidden triple of $(\vec{G},\sigma)$ and thus, not displayed by T by assumption; a contradiction. Therefore, $E = E'$ must hold and thus, $\vec{G}(T,\sigma) = (\vec{G},\sigma)$ which, in particular, implies that $(\vec{G},\sigma)$ is a BMG. □

We can now formulate and prove the main result of this section:

Theorem 4.3. *A vertex colored digraph $(\vec{G},\sigma)$ is a BMG if and only if (i) $(\vec{G},\sigma)$ is sf-colored and (ii) $(\mathcal{R}(\vec{G},\sigma),\mathcal{F}(\vec{G},\sigma))$ is consistent.*

Proof. First suppose that $(\vec{G},\sigma)$ is a BMG. The definition of BMGs together with Obs. 4.1 implies that $(\vec{G},\sigma)$ is sf-colored. Thus, there is a tree (T,σ) that explains $(\vec{G},\sigma)$. By Lemmas 4.1 and 4.12, (T,σ) displays all triples in $\mathcal{R}(\vec{G},\sigma)$ and none of the triples in $\mathcal{F}(\vec{G},\sigma)$. Hence, the pair $(\mathcal{R}(\vec{G},\sigma),\mathcal{F}(\vec{G},\sigma))$ is consistent.

For the converse, suppose that $(\vec{G},\sigma)$ is a sf-colored digraph and that $(\mathcal{R}(\vec{G},\sigma),\mathcal{F}(\vec{G},\sigma))$ is consistent. The latter implies that there is a tree (T,σ) on L displaying all triples in $\mathcal{R}(\vec{G},\sigma)$ and none of the triples in $\mathcal{F}(\vec{G},\sigma)$. Now, we can apply Prop. 4.6 to conclude that $(\vec{G},\sigma)$ is a BMG. ☐

We note that one may extend the set of forbidden triples $\mathcal{F}(\vec{G},\sigma)$ to the set $\mathcal{F}^*(\vec{G},\sigma)$ that consists of *all* triples $ab|b'$ for which $\sigma(a) \neq \sigma(b) = \sigma(b')$ and $(a,b') \in E(\vec{G})$. This may be useful for practical computations. By definition, it holds $\mathcal{F}(\vec{G},\sigma) \subseteq \mathcal{F}^*(\vec{G},\sigma)$.

Corollary 4.7. *A vertex colored digraph $(\vec{G},\sigma)$ is a BMG if and only if (i) $(\vec{G},\sigma)$ is sf-colored and (ii) $(\mathcal{R}(\vec{G},\sigma),\mathcal{F}^*(\vec{G},\sigma))$ is consistent.*

Proof. If $(\vec{G},\sigma)$ satisfies (i) and (ii), then $\mathcal{F}(\vec{G},\sigma) \subseteq \mathcal{F}^*(\vec{G},\sigma)$ implies that $(\mathcal{R}(\vec{G},\sigma),\mathcal{F}(\vec{G},\sigma))$ is consistent. By Thm. 4.3, $(\vec{G},\sigma)$ is a BMG. Now assume that $(\vec{G},\sigma)$ is a BMG. Hence, by Thm. 4.3, $(\vec{G},\sigma)$ is sf-colored and $(\mathcal{R}(\vec{G},\sigma),\mathcal{F}(\vec{G},\sigma))$ is consistent. Hence, there is a tree (T,σ) that displays, in particular, all triples in $\mathcal{R}(\vec{G},\sigma)$. For each triple $xy|y' \in \mathcal{R}(\vec{G},\sigma)$, we have $\sigma(x) \neq \sigma(y) = \sigma(y')$ and $(x,y) \in E(\vec{G})$, $(x,y') \notin E(\vec{G})$. This is precisely the definition of the triples $xy'|y \in F := \mathcal{F}^*(\vec{G},\sigma) \setminus \mathcal{F}(\vec{G},\sigma)$. Hence, none of the triples in F is displayed by T. In summary, (T,σ) displays all triples in $\mathcal{R}(\vec{G},\sigma)$ and none of the triples in $\mathcal{F}(\vec{G},\sigma)$ and F and, therefore, also none of the triples in $\mathcal{F}^*(\vec{G},\sigma)$. Consequently, $(\mathcal{R}(\vec{G},\sigma),\mathcal{F}^*(\vec{G},\sigma))$ is consistent. ☐

As argued in the proof of Cor. 4.7, the information provided by triples in $F := \mathcal{F}^*(\vec{G},\sigma) \setminus \mathcal{F}(\vec{G},\sigma)$ is entirely covered by the set of informative triples $\mathcal{R}(\vec{G},\sigma)$. In the following, we will therefore use the characterization of BMGs as in Thm. 4.3.

In order to use the `MTT` algorithm [128] to recognize BMGs $(\vec{G},\sigma)$, we show for completeness that the set of informative and forbidden triples already determines $V(\vec{G})$ except for trivial cases.

Lemma 4.14. *Let $(\vec{G},\sigma)$ be a sf-colored digraph, $V(\vec{G}) \neq \emptyset$ and $L' := \bigcup_{t \in \mathcal{R}(\vec{G},\sigma) \cup \mathcal{F}(\vec{G},\sigma)} L(t)$. Then the following statements are equivalent:*

1. *$L' = V(\vec{G})$.*
2. *$\mathcal{R}(\vec{G},\sigma) \cup \mathcal{F}(\vec{G},\sigma) \neq \emptyset$.*
3. *$(\vec{G},\sigma)$ is ℓ-colored with $\ell \geq 2$ and contains two vertices of the same color.*

Otherwise, $(\vec{G},\sigma)$ is a BMG that is explained by any tree (T,σ) on $V(\vec{G})$.

Proof. The fact that $L' = V(\vec{G}) \neq \emptyset$ immediately implies that $\mathcal{R}(\vec{G},\sigma) \cup \mathcal{F}(\vec{G},\sigma)$ must not be empty. Hence, (1) implies (2).

Suppose Condition (2) is satisfied. Since all triples $xy|y' \in \mathcal{R}(\vec{G},\sigma) \cup \mathcal{F}(\vec{G},\sigma)$ satisfy $\sigma(x) \neq \sigma(y) = \sigma(y')$, Condition (3) must be satisfied. Hence, (2) implies (3).

Suppose Condition (3) is satisfied. Hence, there are two vertices of the same color r and there must be a vertex $x \in V(\vec{G})$ with $\sigma(x) \neq r$. Since $(\vec{G}, \sigma)$ is sf-colored, there is a vertex $y \in V(\vec{G})$ of color r such that $(x, y) \in E(\vec{G})$. Now let $y' \in V(\vec{G})$, $y' \neq y$ be one of the additional vertices of color $\sigma(y') = r$. If $(x, y') \notin E(\vec{G})$ then $xy|y' \in \mathcal{R}(\vec{G}, \sigma)$ and, otherwise, if $(x, y') \in E(\vec{G})$ then $xy|y' \in \mathcal{F}(\vec{G}, \sigma)$. In summary, every vertex x of $(\vec{G}, \sigma)$ is part of some informative or forbidden triple and thus, $L' = V(\vec{G})$. Hence, (3) implies (1).

Finally, suppose that none of the equivalent statements (1), (2), and (3) holds. Then $(\vec{G}, \sigma)$ is either 1-colored and thus, does not contain any arc, or $|V(\vec{G})|$-colored in which case $(\vec{G}, \sigma)$ is a complete digraph. In both cases, the tree topology of (T, σ) does not matter. □

It is straightforward to test whether a vertex colored digraph $(\vec{G}, \sigma)$ is sf-colored in $O(|E(\vec{G})|)$ time. Moreover, `MTT` [128] accomplishes the consistency check of $(\mathcal{R}(\vec{G}, \sigma), \mathcal{F}(\vec{G}, \sigma))$ and the construction of a corresponding tree in polynomial time (cf. Thm. 2.1). If $\mathcal{R}(\vec{G}, \sigma) \cup \mathcal{F}(\vec{G}, \sigma) = \emptyset$, $(\vec{G}, \sigma)$ is a BMG. Otherwise, Lemma 4.14 implies that every vertex in the sf-colored digraph $(\vec{G}, \sigma)$ appears in an informative and/or a forbidden triple. Together with Prop. 4.6 and Thm. 4.3, this also implies that BMGs can be recognized in polynomial time (cf. Cor. 4.6).

We summarize the characterizations of BMGs in Thm. 4.2 and 4.3 in the following

Proposition 4.7. *Let $(\vec{G}, \sigma)$ be a properly colored digraph with vertex set L. Then the following three statements are equivalent:*

1. *$(\vec{G}, \sigma)$ is a BMG.*
2. *$\mathcal{R}(\vec{G}, \sigma)$ is consistent and $\vec{G}(\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma), L), \sigma) = (\vec{G}, \sigma)$.*
3. *$(\vec{G}, \sigma)$ is sf-colored and $(\mathcal{R}(\vec{G}, \sigma), \mathcal{F}(\vec{G}, \sigma))$ is consistent.*

In this case, $(\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma), L), \sigma)$ is the unique least resolved tree for $(\vec{G}, \sigma)$, and a leaf-colored tree (T, σ) on L explains $(\vec{G}, \sigma)$ if and only if it agrees with $(\mathcal{R}(\vec{G}, \sigma), \mathcal{F}(\vec{G}, \sigma))$.

4.3 BINARY-EXPLAINABLE BEST MATCH GRAPHS

Phylogenetic trees are often binary. Multifurcations are in many cases – but not always – the consequence of insufficient data [73, 257], see also Sec. 3.4 for some discussion on the topic. It is therefore of practical interest to consider BMGs that can be explained by a binary tree.

Definition 4.9. *A properly colored digraph $(\vec{G}, \sigma)$ is a* binary-explainable best match graph (beBMG) *if there is a binary tree T such that $\vec{G}(T, \sigma) = (\vec{G}, \sigma)$.*

In Sec. 4.3.1, we introduce and characterize the *hourglass* as the forbidden induced subgraph for the subclass of beBMGs among BMGs in general. The results of this section previously appeared in [264]. Secs. 4.3.2–4.3.4 are based on [260]. Sec. 4.3.2 is concerned with a constructive algorithm for beBMG recognition that runs in near-cubic time $\tilde{O}(|V|^3)$. It produces a unique tree,

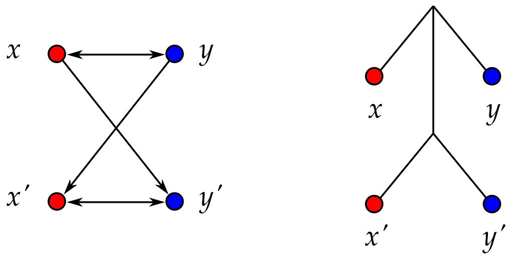


Fig. 13. The hourglass on the left-hand side is explained by the leaf-colored tree on the right-hand side.

the *binary-refinable tree* (BRT) of a beBMG. The BRT has several interesting properties that are studied in detail in Sec. 4.3.3. Simulated data are used in Sec. 4.3.4 to show that BRTs are much better resolved than the least resolved trees of BMGs.

4.3.1 *Hourglasses*

In this section, we will derive a structural characterization of binary-explainable BMGs. To this end, we introduce an additional motif that may occur in vertex-colored digraphs, and that will turn out to be the characteristic forbidden induced subgraph of this subclass of BMGs.

Definition 4.10 (Hourglass). *An* hourglass *in a properly vertex-colored digraph* $(\vec{G},\sigma)$, *denoted by* $[xy \rotatebox[origin=c]{90}{$\bowtie$} x'y']$, *is a subgraph* $(\vec{G}[Q],\sigma_{|Q})$ *induced by a set of four pairwise distinct vertices* $Q=\{x,x',y,y'\}\subseteq V(\vec{G})$ *such that*

(i) $\sigma(x)=\sigma(x')\neq\sigma(y)=\sigma(y')$,

(ii) $(x,y),(y,x)$ *and* $(x',y'),(y',x')$ *are arcs in* $\vec{G}$,

(iii) $(x,y'),(y,x')\in E(\vec{G})$, *and*

(iv) $(y',x),(x',y)\notin E(\vec{G})$.

Note that Condition (i) rules out arcs between x,x' and y,y', respectively, i.e., the only arcs in an hourglass are the ones specified by Conditions (ii) and (iii). The definition of hourglasses is illustrated in Fig. 13, together with the tree that explains a BMG which itself is an hourglass. Hence, we have

Observation 4.5. *Every hourglass is a BMG since it can be explained by a tree as shown in Fig. 13.*

Definition 4.11. *A BMG* $(\vec{G},\sigma)$ *is* hourglass-free *if it does not contain an hourglass as an induced subgraph.*

We first show that hourglasses cannot appear in a BMG that can be explained by a binary tree.

Lemma 4.15. *If* $(\vec{G},\sigma)$ *is a BMG containing the hourglass* $[xy \rotatebox[origin=c]{90}{$\bowtie$} x'y']$, *then every tree* (T,σ) *that explains* $(\vec{G},\sigma)$ *contains a vertex* $u\in V^0(T)$ *with three distinct children* v_1, v_2, *and* v_3 *such that* $x\preceq_T v_1$, $\operatorname{lca}_T(x',y')\preceq_T v_2$ *and* $y\preceq_T v_3$.

Proof. By assumption, $(x,y),(y,x)$ and $(x',y'),(y',x')$ are arcs in $\vec{G}$, $(x,y'),(y,x') \in E(\vec{G})$, and $(y',x),(x',y) \notin E(\vec{G})$. By Lemma 4.1, the informative triples $x'y'|x$ and $x'y'|y$ thus must be displayed by every tree (T,σ) that explains $(\vec{G},\sigma)$. Thus $u_{x'y'} := \operatorname{lca}_T(x',y') \prec_T u_x := \operatorname{lca}_T(x,u_{x'y'})$ and $u_{x'y'} \prec_T u_y := \operatorname{lca}_T(y,u_{x'y'})$. Furthermore, u_x and u_y are both ancestors of $u_{x'y'}$ and thus comparable w.r.t. $\preceq_T$. If $u_x \prec_T u_y$, then $\operatorname{lca}_T(x,y') \prec_T \operatorname{lca}_T(x,y)$ which implies that (x,y) cannot be an arc in $\vec{G}$; a contradiction. By similar arguments, $u_y \prec_T u_x$ is not possible and therefore, $u_x = u_y =: u$.

Since $u_{x'y'} \prec_T u$, there are two distinct children $v_1, v_2 \in \operatorname{child}_T(u)$ of u such that $x \preceq_T v_1$ and $u_{x'y'} \preceq_T v_2$. Clearly, $y \notin L(T(v_2))$ since $\operatorname{lca}_T(y,u_{x'y'}) = u \succ_T v_2$. We also have $y \notin L(T(v_1))$ since $y \in L(T(v_1))$ would imply $\operatorname{lca}_T(x,y) \preceq_T v_1 \prec_T u = \operatorname{lca}_T(x,u_{x'y'}) = \operatorname{lca}_T(x,y')$, contradicting $(x,y') \in E(\vec{G})$. Together with $y \in L(T(u))$, this implies the existence of a vertex $v_3 \in \operatorname{child}_T(u)$ such that $v_3 \notin \{v_1,v_2\}$ and $y \preceq_T v_3$. ☐

As an immediate consequence, we obtain

Corollary 4.8. *If a BMG $(\vec{G},\sigma)$ is binary-explainable, then it is hourglass-free.*

As we shall see, the converse of Cor. 4.8 is also true. In order to show this, we will need the following technical result.

Lemma 4.16. *Let $(\vec{G},\sigma)$ be a BMG explained by (T,σ). Then $(\vec{G},\sigma)$ has an hourglass $[xy \mathrel{\nearrow\mkern-18mu\searrow} x'y']$ as an induced subgraph if and only if there is a vertex $u \in V^0(T)$ with distinct children v_1, v_2, and v_3 and two distinct colors r and s satisfying*

1. *$r \in \sigma(L(T(v_1)))$, $r,s \in \sigma(L(T(v_2)))$, and $s \in \sigma(L(T(v_3)))$, and*
2. *$s \notin \sigma(L(T(v_1)))$, and $r \notin \sigma(L(T(v_3)))$.*

Proof. First assume that $(\vec{G},\sigma)$ contains the hourglass $[xy \mathrel{\nearrow\mkern-18mu\searrow} x'y']$ as an induced subgraph. Then by Lemma 4.15, (T,σ) contains a vertex $u \in V^0(T)$ with three distinct children v_1, v_2, and v_3 such that $x \preceq_T v_1$, $\operatorname{lca}_T(x',y') \preceq_T v_2$ and $y \preceq_T v_3$. Putting $r := \sigma(x) = \sigma(x')$ and $s := \sigma(y) = \sigma(y')$ immediately implies Condition (1). Now, assume for contradiction that Condition (2) is violated and thus $s \in \sigma(L(T(v_1)))$ or $r \in \sigma(L(T(v_3)))$. If $s \in \sigma(L(T(v_1)))$, then there is a leaf $y'' \prec_T v_1$ with $\sigma(y'') = s$. In this case, however, $\operatorname{lca}_T(x,y'') \preceq_T v_1 \prec_T u = \operatorname{lca}_T(x,y')$ implies that (x,y') cannot be an arc in $(\vec{G},\sigma)$; a contradiction to $[xy \mathrel{\nearrow\mkern-18mu\searrow} x'y']$ being an hourglass. By similar arguments, $r \in \sigma(L(T(v_3)))$ is not possible. Therefore, Condition (2) must be satisfied.

Now assume that there is a vertex $u \in V^0(T)$ with pairwise distinct children v_1, v_2, and v_3 and two distinct colors r and s satisfying Conditions (1) and (2). It is now straightforward to see that $(\vec{G},\sigma)$ contains an hourglass: Condition (1) immediately implies the existence of vertices $x \in L[r] \cap L(T(v_1))$ and $y \in L[s] \cap L(T(v_3))$. Moreover, $r,s \in \sigma(L(T(v_2)))$ together with Lemma 4.3 imply that there is are arcs $(x',y'),(y',x')$ in $(\vec{G},\sigma)$ with $x' \in L[r] \cap L(T(v_2))$ and $y' \in L[s] \cap L(T(v_2))$. Clearly, the vertices in $\{x,x',y,y'\}$ are pairwise distinct. By Condition (2) and the location of the four leaves, we obtain the arcs (x,y'), (x,y), (y,x'), and (y,x). Since $T(v_2)$ contains both colors r and s, we can furthermore conclude that (x',y) and (y',x) are not arcs in $(\vec{G},\sigma)$. In summary, the subgraph of $(\vec{G},\sigma)$ induced by the set $\{x,x',y,y'\}$ is an hourglass $[xy \mathrel{\nearrow\mkern-18mu\searrow} x'y']$. ☐

In the following, a tree (T,σ) is called *refinable* if there is a proper refinement (T',σ) of (T,σ), i.e., $T \leq T'$ and $T \neq T'$, such that $\vec{G}(T',\sigma) = \vec{G}(T,\sigma)$. Otherwise, (T,σ) is *non-refinable*. An inner vertex of a tree is *non-refinable* if it cannot be refined without changing the best match graph explained by the tree.

Clearly, for every BMG $(\vec{G},\sigma)$, there is a tree that has the maximum number of vertices among all trees that explain $(\vec{G},\sigma)$ and thus, a tree that cannot be further resolved. Hence, every BMG can be explained by a non-refinable tree. We will need the following useful property of non-refinable vertices:

Lemma 4.17. *Let $(\vec{G},\sigma)$ be a BMG explained by a tree (T,σ), and let $u \in V^0(T)$ be a non-refinable vertex of (T,σ). Then, for any proper subset $C \subsetneq \mathsf{child}_T(u)$ with $|C| \geq 2$, there are two distinct vertices $v, v' \in C$, a vertex $v'' \in \mathsf{child}_T(u) \setminus C$, and two vertices $a \preceq_T v$ and $b \preceq_T v'$ such that $(a,b) \in E(\vec{G})$ and $\sigma(b) \in \sigma(L(T(v'')))$.*

Proof. First note that the statement is trivially true if u is binary, since then there is no proper subset $C \subsetneq \mathsf{child}_T(u)$ such that $|C| \geq 2$. Thus, assume $|\mathsf{child}_T(u)| \geq 3$ in the following.

We refine (T,σ) at vertex u as follows: Take an arbitrary subset $C \subsetneq \mathsf{child}_T(u)$ such that $|C| \geq 2$ (which exists since $|\mathsf{child}_T(u)| \geq 3$) and place all vertices in C as the children of a new vertex w, and connect w as a child of u. Since u is a non-refinable vertex of (T,σ), this refinement leads to a tree (T',σ) that does not explain $(\vec{G},\sigma)$, and therefore, the inner edge uw must be non-redundant w.r.t. $\vec{G}(T',\sigma)$. By Lemma 4.6, there must be an arc (a,b) in $\vec{G}(T',\sigma)$ such that $\mathrm{lca}_{T'}(a,b) = w$ and $\sigma(b) \in \sigma(L(T'(u)) \setminus L(T'(w)))$. In particular, $\mathrm{lca}_{T'}(a,b) = w$ implies that $a \preceq_T v$ and $b \preceq_T v'$ for two distinct vertices $v, v' \in \mathsf{child}_{T'}(w) = C$. Note that (T,σ) can be obtained from (T',σ) by contraction of the edge uw. Hence, we can apply Lemma 4.7 to conclude that $\vec{G}(T',\sigma) \subseteq (\vec{G},\sigma)$. Therefore, $(a,b) \in E(\vec{G})$. Taking the latter arguments together, for any subset $C \subsetneq \mathsf{child}_T(u)$ with $|C| \geq 2$, there are vertices $a \preceq_T v$ and $b \preceq_T v'$ with distinct $v, v' \in C$ such that $(a,b) \in E(\vec{G})$ and $\sigma(b) \in \sigma(L(T(v'')))$ for some $v'' \in \mathsf{child}_T(u) \setminus C$. ☐

We now show that hourglasses are the forbidden induced subgraph characterizing BMGs that can be explained by binary trees.

Proposition 4.8. *A BMG $(\vec{G},\sigma)$ can be explained by a binary tree if and only if it is hourglass-free.*

Proof. By Cor. 4.8, every binary-explainable BMG $(\vec{G},\sigma)$ must be hourglass-free. To prove the converse, we assume, for contradiction, that $(\vec{G},\sigma)$ is hourglass-free and cannot be explained by any binary tree. Then there is a non-refinable non-binary tree (T,σ) that explains $(\vec{G},\sigma)$. By construction, furthermore, T contains a non-binary vertex $u \in V^0(T)$, which by assumption is non-refinable.

The key device for our proof are pairs $(\mathcal{M},\mathcal{N})$ where $\mathcal{M} := \{v_1,\dots,v_k\}$ is an ordered set of $k \geq 2$ pairwise distinct children of u and $\mathcal{N} := \{c_1,\dots,c_{k-1}\}$ is an ordered set of $k-1$ pairwise distinct colors. We call $(\mathcal{M},\mathcal{N})$ an *hourglass-free pair (hf-pair) of order k* for u if the following conditions are satisfied:
(i) For all $c_i \in \mathcal{N}$ we have $c_i \in \sigma(L(T(v_j)))$, $i \leq j \leq k-1$,
(ii) For all $c_i \in \mathcal{N}$ we have $c_i \notin \sigma(L(T(v_j)))$, $1 \leq j < i$, and
(iii) $\mathcal{N} \subseteq \sigma(L(T(v_k)))$.
If $(\mathcal{M},\mathcal{N})$ is an hf-pair of order k, then Condition (i) implies by construction that $\mathcal{N} \subseteq \sigma(L(T(v_{k-1})))$. Therefore, $(\mathcal{M}' = (v_1,\dots,v_k,v_{k-1}),\mathcal{N})$ is also an hf-pair where $\mathcal{M}'$ is obtained from $\mathcal{M}$ by exchanging the positions of its last two elements. Hf-pairs and the following arguments are illustrated in Fig. 14. In order to obtain the desired contradiction, we show by induction that the children of the non-binary, non-refinable vertex u harbor hf-pairs of arbitrary large order k.

Base case. There is an hf-pair $(\mathcal{M}, \mathcal{N})$ of order 2 for u.
Proof of Claim. Consider an arbitrary subset $\{v, v'\} \subsetneq \mathsf{child}_T(u)$ consisting of two distinct children v and v' of the non-binary vertex u. By Lemma 4.17 and since u is non-refinable, there are vertices $a \preceq_T v$ and $b \preceq_T v'$ such that w.l.o.g. $(a, b) \in E(\vec{G})$ and $\sigma(b) \in \sigma(L(T(v'')))$ for some $v'' \in \mathsf{child}_T(u) \setminus \{v, v'\}$. The latter implies that there is a vertex $b' \preceq_T v''$ of color $\sigma(b)$. Clearly, b and b' are distinct and the color $\sigma(b)$ is also present in the subtree $T(v')$. Thus we can set $\mathcal{M} := (v_1 := v', v_2 := v'')$ and $\mathcal{N} := (c_1 := \sigma(b))$. It is an easy task to verify that $(\mathcal{M}, \mathcal{N})$ satisfies Conditions (i)–(iii). $\diamond$

Induction step. The existence of an hf-pair of order k implies the existence of an hf-pair of order $k+1$ for u.
Proof of Claim. Let $(\mathcal{M} = (v_1, \dots, v_k), \mathcal{N} = (c_1, \dots, c_{k-1}))$ be an hf-pair, and consider the set $\{v_{k-1}, v_k\} \subsetneq \mathsf{child}_T(u)$. By Lemma 4.17 and since u is non-refinable, there are again vertices $a \preceq_T v$ and $b \preceq_T v'$ for distinct $v, v' \in \{v_{k-1}, v_k\}$ such that $(a, b) \in E(\vec{G})$ and $\sigma(b) \in \sigma(L(T(v'')))$ for some $v'' \in \mathsf{child}_T(u) \setminus \{v_{k-1}, v_k\}$. We can assume w.l.o.g. that $a \preceq_T v = v_{k-1}$ and $b \preceq_T v' = v_k$ since otherwise we can simply swap v_{k-1} and v_k in the ordered set $\mathcal{M}$ as argued above. Since (a, b) is an arc in $(\vec{G}, \sigma)$ and $\mathrm{lca}_T(a, b) = u$, the color $\sigma(b)$ cannot be present in the subtree $T(v_{k-1})$. Since $\mathcal{N} \subseteq \sigma(L(T(v_{k-1})))$ and $\sigma(b) \notin \sigma(L(T(v_{k-1})))$, we conclude that $\sigma(b) \notin \mathcal{N}$.

We continue to show that v'' is distinct from all elements in $\mathcal{M}$. Clearly, in the case $k = 2$, v'' is distinct from all elements in $\mathcal{M} = \{v_1, v_2\} = \{v, v'\}$ by construction. Now let $k > 2$ and assume, for contradiction, that there is a vertex $v_j \in \{v_1, \dots, v_{k-2}\}$ such that $\sigma(b) \in \sigma(L(T(v_j)))$. In this case, $j < k-1$ and Condition (ii) imply that $c_{k-1} \notin \sigma(L(T(v_j)))$. In addition, we have $c_{k-1} \in \sigma(L(T(v_{k-1})))$ and $c_{k-1} \in \sigma(L(T(v_k)))$ by Conditions (i) and (iii), respectively. Recall that $v' = v_k$. In summary, we obtain three distinct vertices v_j, v_k, v_{k-1} and two distinct colors $\sigma(b)$ and c_{k-1} satisfying Conditions (1) and (2) in Lemma 4.16, which implies that $(\vec{G}, \sigma)$ contains an hourglass; a contradiction. Hence, $\sigma(b) \notin \sigma(L(T(v_j)))$ for all $j \in \{1, \dots, k-2\}$. This implies that v'' is distinct from $v_1, \dots, v_{k-2}$. Moreover, by construction, v'' is distinct from v_{k-1} and v_k. In summary, v'' is therefore distinct from all elements in $\mathcal{M}$.

Consider now the pair $(\mathcal{M}' := (v_1, \dots, v_k, v_{k+1} := v''), \mathcal{N}' := (c_1, \dots, c_{k-1}, c_k := \sigma(b)))$. Since $(\mathcal{M}, \mathcal{N})$ is an hf-pair, and since, by construction, $c_k = \sigma(b) \notin \sigma(L(T(v_j)))$ for $1 \leq j \leq k-1$ and $c_k = \sigma(b) \in \sigma(L(T(v_k)))$, we can immediately conclude that Conditions (i) and (ii) are satisfied for $(\mathcal{M}', \mathcal{N}')$. It remains to show that Condition (iii) is satisfied as well, i.e., $c_i \in \sigma(L(T(v_{k+1})))$ for all $1 \leq i \leq k$. By construction, we have $c_k \in \sigma(L(T(v_{k+1})))$. Now assume that $c_i \notin \sigma(L(T(v_{k+1})))$ for some $1 \leq i \leq k-1$. We have $c_i \in \sigma(L(T(v_{k-1})))$ and $c_i, c_k \in \sigma(L(T(v_k)))$ by Condition (i), and $c_k \notin \sigma(L(T(v_{k-1})))$ by Condition (ii). Taken together, we obtain three distinct vertices v_{k-1}, v_k, v_{k+1} and two distinct colors c_i and c_k satisfying Conditions (1) and (2) in Lemma 4.16, which implies that $(\vec{G}, \sigma)$ contains an hourglass; a contradiction. Therefore, Condition (iii) must be satisfied as well, and $(\mathcal{M}', \mathcal{N}')$ is an hf-pair of order $k+1$. $\diamond$

Repeated application of the induction step implies that children of a non-refinable non-binary vertex u in a non-refinable tree (T, σ) explaining an hourglass-free BMG harbor an hf-pair of arbitrary order. This is of course impossible since $\vec{G}$ is finite, i.e, no such vertex u can exist. Therefore, every hourglass-free BMG $(\vec{G}, \sigma)$ can be explained by a binary tree. $\square$

Prop. 4.8 gives rise to a procedure for determining whether a BMG $(\vec{G}, \sigma)$ can be explained by a binary tree. We simply need to check whether $(\vec{G}, \sigma)$ is hourglass-free, a task that can be done trivially in $O(|E(\vec{G})|^2)$ time by checking, for all pairs of bidirectional arcs $(a, b), (b, a)$ and $(a', b'), (b', a')$ in con-

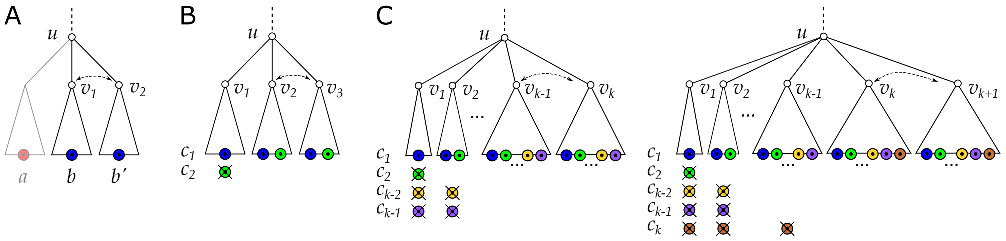


Fig. 14. Illustration of the induction argument in the proof of Prop. 4.8. (A) Base case: an hourglass-free pair (hf-pair) $(\mathcal{M} = \{v_1, v_2\}, \mathcal{N} = \{\sigma(b)\})$ of order 2. Note that vertex a is only required to show the existence of $(\mathcal{M}, \mathcal{N})$. (B) An hf-pair $(\mathcal{M} = \{v_1, v_2, v_3\}, \mathcal{N} = \{c_1, c_2\})$ of order 3. (C) Induction step: The existence of an hf-pair of order k implies the existence of an hf-pair of order $k+1$, and thus, an infinite number of children of u. This gives the desired contradiction in the proof of Prop. 4.8. The dashed arrow indicates the last two elements in the ordered set $\mathcal{M}$ of an hf-pair $(\mathcal{M}, \mathcal{N})$ are interchangeable.

stant time, whether or not they induce an hourglass $[ab \mathrel{\searrow\!\!\!\!\!\!\nearrow} a'b']$ or $[a'b' \mathrel{\searrow\!\!\!\!\!\!\nearrow} ab]$. Hence, we obtain

Corollary 4.9. *It can be decided in polynomial time whether a BMG $(\vec{G}, \sigma)$ can be explained by a binary tree.*

We shall see, however, in the next section that this can be achieved faster using a certain extended set of informative triples.

4.3.2 *Binary Trees Explaining a BMG in Near Cubic Time*

The characterization of binary-explainable BMGs given in Prop. 4.8 is not constructive, i.e., it does not answer the question of how to construct a binary tree explaining a given BMG, if one exists. In this section, we derive an efficient algorithm for this problem, and show that it can be expressed as a consistency problem involving a certain set of triples obtained from the informative and the forbidden triples. It is therefore related to the MOST RESOLVED COMPATIBLE TREE and FORBIDDEN TRIPLES (RESTRICTED TO BINARY TREES) problems, both of which are NP-complete [35]. However, we have already seen that it can be decided in $O(|E(\vec{G})|^2)$ time whether a BMG is binary-explainable (cf. Cor. 4.9). In particular, this time complexity will be reduced further.

We start with a few technical results on the structure of the triples sets $\mathcal{R}(\vec{G}, \sigma)$ and $\mathcal{F}(\vec{G}, \sigma)$. To this end, recall that, by Obs. 4.4, forbidden triples always come in pairs of the form $ab|b'$ and $ab'|b$.

Lemma 4.18. *Let $(\vec{G}, \sigma)$ be explained by a binary tree (T, σ). If $ab|b' \in \mathcal{F}(\vec{G}, \sigma)$ with $\sigma(b) = \sigma(b')$, then (T, σ) displays the triple $bb'|a$.*

Proof. Suppose that $ab|b' \in \mathcal{F}(\vec{G}, \sigma)$ with $\sigma(b) = \sigma(b')$, and recall that a, b, b' must be pairwise distinct. By Obs. 4.4, we have $ab'|b \in \mathcal{F}(\vec{G}, \sigma)$. By Prop. 4.7 and since (T, σ) explains $(\vec{G}, \sigma)$, (T, σ) displays none of the two forbidden triples $ab|b'$ and $ab'|b$. However, the fact that (T, σ) is binary implies that exactly one triple on $\{a, b, b'\}$ must be displayed, of which only $bb'|a$ remains. □

Lemma 4.18 implies that we can infer a set of additional triples that would be required for a binary tree to explain a vertex-colored digraph $(\vec{G},\sigma)$. This motivates the definition of an extended informative triple set

$$\mathcal{R}^{\mathrm{B}}(\vec{G},\sigma) := \mathcal{R}(\vec{G},\sigma) \cup \{bb'|a\colon ab|b' \in \mathcal{F}(\vec{G},\sigma) \text{ and } \sigma(b)=\sigma(b')\}. \qquad (7)$$

Since informative and forbidden triples are defined by the presence and absence of certain arcs in a vertex-colored digraph, this leads to the following

Observation 4.6. *Let $(\vec{G},\sigma)$ be a vertex-colored digraph and $L' \subseteq V(\vec{G})$. Then $R(\vec{G},\sigma)_{|L'} = R(\vec{G}[L'],\sigma_{|L'})$ holds for any $R \in \{\mathcal{R},\mathcal{F},\mathcal{R}^B\}$.*

Lemma 4.19. *If (T,σ) is a binary tree explaining the BMG $(\vec{G},\sigma)$, then (T,σ) displays $\mathcal{R}^B(\vec{G},\sigma)$.*

Proof. Let (T,σ) be a binary tree that explains $(\vec{G},\sigma)$. By Prop. 4.7, $(\vec{G},\sigma)$ displays all informative triples $\mathcal{R}(\vec{G},\sigma)$. Now let $bb'|a \in \mathcal{R}^{\mathrm{B}}(\vec{G},\sigma) \setminus \mathcal{R}(\vec{G},\sigma)$. Hence, by definition and Obs. 4.4, $ab|b'$ and $ab'|b$ are forbidden triples for $(\vec{G},\sigma)$. This together with Lemma 4.18 and the fact that (T,σ) is binary implies that $bb'|a$ is displayed by (T,σ). In summary, therefore, (T,σ) displays all triples in $\mathcal{R}^{\mathrm{B}}(\vec{G},\sigma)$. □

Lemma 4.20. *Let $(\vec{G},\sigma)$ be an sf-colored digraph with vertex set L. Every tree on L that displays $\mathcal{R}^B(\vec{G},\sigma)$ explains $(\vec{G},\sigma)$.*

Proof. Suppose that a tree (T,σ) on L displays $\mathcal{R}^{\mathrm{B}}(\vec{G},\sigma)$ and thus, in particular, $\mathcal{R}(\vec{G},\sigma)$. Now suppose $ab|b' \in \mathcal{F}(\vec{G},\sigma)$ with $\sigma(b)=\sigma(b')$ is a forbidden triple for $(\vec{G},\sigma)$ and hence, $bb'|a \in \mathcal{R}^{\mathrm{B}}(\vec{G},\sigma)$. Clearly, (T,σ) displays at most one of the three possible triples on $\{a,b,b'\}$. Taken together, the latter arguments imply that (T,σ) does not display $ab|b'$. In summary, (T,σ) displays all triples in $\mathcal{R}(\vec{G},\sigma)$ and none of the triples in $\mathcal{F}(\vec{G},\sigma)$ and thus, $(\mathcal{R}(\vec{G},\sigma),\mathcal{F}(\vec{G},\sigma))$ is consistent. Therefore and since $(\vec{G},\sigma)$ is sf-colored by assumption, we can apply Prop. 4.7 to conclude that the tree (T,σ) on L explains the BMG $(\vec{G},\sigma)$. □

Using Lemmas 4.19 and 4.20, it can be shown that consistency of $\mathcal{R}^{\mathrm{B}}(\vec{G},\sigma)$ is sufficient for an sf-colored digraph $(\vec{G},\sigma)$ to be binary-explainable.

Theorem 4.4. *A properly vertex-colored digraph $(\vec{G},\sigma)$ with vertex set L is binary-explainable if and only if (i) $(\vec{G},\sigma)$ is sf-colored, and (ii) $\mathcal{R}^B := \mathcal{R}^B(\vec{G},\sigma)$ is consistent. In this case, the BMG $(\vec{G},\sigma)$ is explained by every refinement of the tree $(\mathrm{Aho}(\mathcal{R}^B,L),\sigma)$.*

Proof. First suppose that $(\vec{G},\sigma)$ is sf-colored and that $\mathcal{R}^{\mathrm{B}}$ is consistent. Therefore, the tree $T := \mathrm{Aho}(\mathcal{R}^{\mathrm{B}},L)$ exists. By correctness of `BUILD` [7], T displays all triples in $\mathcal{R}^{\mathrm{B}}$. Clearly, every refinement T' of T also displays $\mathcal{R}^{\mathrm{B}}$. Hence, for every refinement T' of T (including T itself), we can apply Lemma 4.20 to conclude that (T',σ) explains $(\vec{G},\sigma)$. In particular, $(\vec{G},\sigma)$ is a BMG. Since there always exists a binary refinement of T, the latter arguments imply that $(\vec{G},\sigma)$ is binary-explainable.

Now suppose that $(\vec{G},\sigma)$ can be explained by a binary tree (T,σ), and note that $(\vec{G},\sigma)$ is a BMG in this case. By Prop. 4.7, $(\vec{G},\sigma)$ is sf-colored. Moreover, the binary tree (T,σ) displays $\mathcal{R}^{\mathrm{B}}$ as a consequence of Lemma 4.19. Therefore, $\mathcal{R}^{\mathrm{B}}$ must be consistent. □

Algorithm 2: Construction of a binary tree explaining $(\vec{G},\sigma)$.

```
Input: A properly vertex-colored digraph $(\vec{G},\sigma)$ with vertex set $L$.
Output: Binary tree $(T,\sigma)$ explaining $(\vec{G},\sigma)$ if one exists.
if $(\vec{G},\sigma)$ is not sf-colored then
    exit false
construct the extended triple set $\mathcal{R}^B := \mathcal{R}^B(\vec{G},\sigma)$
$T \leftarrow$ Aho$(\mathcal{R}^B, L)$
if $T$ is a tree then
    construct an arbitrary binary refinement $T'$ of $T$
    return $(T',\sigma)$
else
    exit false
```

Thm. 4.4 implies that the problem of determining whether an sf-colored digraph $(\vec{G},\sigma)$ is binary-explainable can be reduced to a triple consistency problem. More precisely, it establishes the correctness of Alg. 2, which in turn relies on the construction of $\mathrm{Aho}(\mathcal{R}^B, L)$. The latter can be achieved in polynomial time [7]. Making use of the improvements achievable by using dynamic graph data structures [71, 139], we obtain the following performance bound:

Corollary 4.10. *There exists an $O(|L|^3 \log^2 |L|)$-time algorithm that constructs a binary tree explaining a vertex-colored digraph $(\vec{G},\sigma)$ with vertex set L, if and only if such a tree exists.*

Proof. For a vertex-colored digraph $(\vec{G},\sigma)$ with vertex set L it can be decided in $O(|L|^2)$ whether it is sf-colored, i.e., whether it is properly colored and every vertex has an out-neighbor with every other color. The set $\mathcal{R}^B := \mathcal{R}^B(\vec{G},\sigma)$ can easily be constructed in $O(|L|^3)$ using Eqs. (3), (6) and (7) and the number of triples in $\mathcal{R}^B$ is bounded by $O(|L|^3)$. Note that every triple in $\mathcal{R}^B$ is a tree with a constant number of vertices and edges. Thus, the total number M of vertices and edges in $\mathcal{R}^B$ is also in $O(|L|^3)$. The algorithm `BuildST` [71] solves the consistency problem for $\mathcal{R}^B$ and constructs a corresponding (not necessarily binary) tree T in $O(M \log^2 M) = O(|L|^3 \log^2 |L|)$ time [71, Thm. 3]. Finally, we can obtain an arbitrary binary refinement T' of T in $O(|L|)$. Thus there exists a version of Alg. 2 that solves the problem in $O(|L|^3 \log^2 |L|)$ time. ☐

We close this section with a summary of the characterizations of beBMGs among BMGs in general.

Proposition 4.9. *For every BMG $(\vec{G},\sigma)$ explained by a tree (T,σ), the following three statements are equivalent:*

1. *$(\vec{G},\sigma)$ is binary-explainable.*
2. *$(\vec{G},\sigma)$ is hourglass-free.*
3. *There is no vertex $u \in V^0(T)$ with three distinct children v_1, v_2, and v_3 and two distinct colors r and s satisfying*
    a) *$r \in \sigma(L(T(v_1)))$, $r, s \in \sigma(L(T(v_2)))$, and $s \in \sigma(L(T(v_3)))$, and*

b) $s \notin \sigma(L(T(v_1)))$, and $r \notin \sigma(L(T(v_3)))$.

4. *$\mathcal{R}^B(\vec{G},\sigma)$ is consistent.*

Proof. By Prop. 4.8 and Lemma 4.16, Statements (1) to (3) are equivalent. By Obs. 4.1 and Def. 4.2, the BMG $(\vec{G},\sigma)$ is sf-colored. By Thm. 4.4, an sf-colored graph is binary-explainable if and only if $\mathcal{R}^B(\vec{G},\sigma)$ is consistent. It follows that Statements (1) and (4), and thus all of them, are equivalent. ☐

4.3.3 *The Binary-Resolvable Tree*

If a digraph $(\vec{G},\sigma)$ with vertex set L is binary-explainable, Thm. 4.4 implies that $\mathcal{R}^B := \mathcal{R}^B(\vec{G},\sigma)$ is consistent and every refinement of $(\mathrm{Aho}(\mathcal{R}^B, L),\sigma)$ explains $(\vec{G},\sigma)$. In this section, we investigate the properties of this tree in more detail.

Definition 4.12. *The* binary-refinable tree (BRT) *of a binary-explainable BMG $(\vec{G},\sigma)$ with vertex set L is the leaf-colored tree $(\mathrm{Aho}(\mathcal{R}^B(\vec{G},\sigma), L),\sigma)$.*

The BRT is not necessarily a binary tree. However, Thm. 4.4 implies that the BRT as well as each of its binary refinements explain $(\vec{G},\sigma)$. Note that the tree $\mathrm{Aho}(\mathcal{R}^B, L)$ and thus the BRT are well-defined because Thm. 4.4 ensures consistency of $\mathcal{R}^B$ for binary-explainable digraphs, and the Aho tree as produced by `BUILD` is uniquely determined by the set of input triples [7].

Corollary 4.11. *If $(\vec{G},\sigma)$ is a binary-explainable BMG, then its BRT is a refinement of the LRT.*

Proof. Since each BMG has a unique LRT [102, Thm. 8], the BRT of a binary-explainable BMG is necessarily a refinement of the LRT. ☐

Clearly, the BRT is least resolved among the trees that display $\mathcal{R}^B$, i.e., contraction of an arbitrary edge results in a tree that no longer displays all triples in $\mathcal{R}^B$ [270, Prop. 4.1]. Now, we tackle the question whether the BRT is the unique least resolved tree in this sense. In other words, we ask whether every tree that displays $\mathcal{R}^B$ is a refinement of the BRT. As we shall see, this question can be answered in the affirmative.

We will make extensive use of the closure operator for sets of rooted triples. To this end, recall the properties of $\mathrm{cl}(\mathcal{R})$ and Prop. 2.4 given in Sec. 2.3.5.

The following extension of Lemma 4.14 shows that Prop. 2.4 can be applied to the triple set $\mathcal{R}^B(\vec{G},\sigma)$ of an sf-colored digraph $(\vec{G},\sigma)$ with the exception of the two trivial special cases in which either all vertices of $(\vec{G},\sigma)$ are of the same color or of pairwise distinct colors.

Lemma 4.21. *Let $(\vec{G},\sigma)$ be an sf-colored digraph with vertex set $L \neq \emptyset$, $L_{\mathcal{R},\mathcal{F}} := \bigcup_{t\,\in\,\mathcal{R}(\vec{G},\sigma)\,\cup\,\mathcal{F}(\vec{G},\sigma)} L(t)$ and $L_{\mathcal{R}^B} := \bigcup_{t\,\in\,\mathcal{R}^B(\vec{G},\sigma)} L(t)$. Then the following statements are equivalent:*

1. *$L_{\mathcal{R},\mathcal{F}} = L_{\mathcal{R}^B} = L$.*
2. *$\mathcal{R}(\vec{G},\sigma) \cup \mathcal{F}(\vec{G},\sigma) \neq \emptyset$.*
3. *$\mathcal{R}^B(\vec{G},\sigma) \neq \emptyset$.*

4. *$(\vec{G},\sigma)$ is ℓ-colored with $\ell \geq 2$ and contains two vertices of the same color.*

If these statements are not satisfied, then $(\vec{G},\sigma)$ is a BMG that is explained by any tree (T,σ) on L.

Proof. By Lemma 4.14, Statements 2 and 4, and $L_{\mathcal{R},\mathcal{F}} = L$ are equivalent. One easily verifies using Eqs. (3) and (7) that there is a triple on $\{a,b,c\}$ in $\mathcal{R}(\vec{G},\sigma) \cup \mathcal{F}(\vec{G},\sigma)$ if and only if there is a triple on $\{a,b,c\}$ in $\mathcal{R}^{\mathrm{B}}(\vec{G},\sigma)$. Therefore, Statements 2 and 3 are equivalent and we always have $L_{\mathcal{R},\mathcal{F}} = L_{\mathcal{R}^{\mathrm{B}}}$. Thus all statements are equivalent. If the statements are not satisfied, i.e., in particular, Statement (4) is not satisfied, then the vertices in L are all either of the same or of different color. In both cases, $(\vec{G},\sigma)$ is explained by any tree on L. ☐

Lemma 4.21 holds for BMGs since these are sf-colored by Prop. 4.7. The following result is essential for the application of Prop. 2.4 to a triple set $\mathcal{R}^{\mathrm{B}}(\vec{G},\sigma)$.

Lemma 4.22. *Let $(\vec{G},\sigma)$ be a binary-explainable BMG with vertex set L and $\mathcal{R}^B := \mathcal{R}^B(\vec{G},\sigma)$. Then, for any two distinct connected components C and C' of the Aho graph $H := [\mathcal{R}^B, L]$, the subgraph $H[L']$ induced by $L' = V(C) \mathbin{\dot\cup} V(C')$ satisfies $H[L'] = [\mathcal{R}^B{}_{|L'}, L'] = C \mathbin{\dot\cup} C'$.*

Proof. Since $(\vec{G},\sigma)$ is binary-explainable, $\mathcal{R}^{\mathrm{B}}$ is consistent by Thm. 4.4. Thus $H := [\mathcal{R}^{\mathrm{B}}, L]$ contains at least two connected components. If H contains exactly two connected components C and C', the statement trivially holds. Hence, assume that H contains at least three connected components. Let C and C' be two distinct connected components of H, and set $L' := V(C) \cup V(C')$ and $H' := [\mathcal{R}^{\mathrm{B}}{}_{|L'}, L']$. Note, $V(H[L']) = V(H') = L'$, and $H[L'] = C \mathbin{\dot\cup} C'$ is the induced subgraph of H that consists precisely of the two connected components C and C'. From $\mathcal{R}^{\mathrm{B}}{}_{|L'} \subseteq \mathcal{R}^{\mathrm{B}}$ and the construction of H we immediately observe that H' is a subgraph of $H[L']$. Hence, it remains to show that every edge xy in $H[L']$ is also an edge in H'.

To this end, we consider the BRT (T,σ) of $(\vec{G},\sigma)$, which exists since $\mathcal{R}^{\mathrm{B}}$ is consistent and explains $(\vec{G},\sigma)$ by Thm. 4.4. By construction, there is a one-to-one correspondence between the connected components of H and the children of the root ρ of T. Thus, let v and v' be the distinct children of ρ such that $L(T(v)) = V(C)$ and $L(T(v')) = V(C')$ and let xy be an edge in $H[L']$. Since x and y lie in the same connected component of H and $x,y \in L'$, we can assume w.l.o.g. that $x,y \in L(T(v))$. It suffices to show that there is a triple $xy|z \in \mathcal{R}^{\mathrm{B}}$ with $z \in L'$, since in this case, we obtain $xy|z \in \mathcal{R}^{\mathrm{B}}{}_{|L'}$ and thus $xy \in E(H')$.

We assume, for contradiction, that there is no $z \in L'$ with $xy|z \in \mathcal{R}^{\mathrm{B}}$. Then, by construction of H and since xy is an edge therein, $\mathcal{R}^{\mathrm{B}}$ contains a triple $xy|z$ with $z \in L(T(v''))$ for some $v'' \in \mathsf{child}_T(\rho) \setminus \{v,v'\}$ and a connected component C'' of H with $V(C'') = L(T(v''))$. By Eqs. (3) and (7), there are exactly two cases for such a triple:

(a) $xy|z = ab|b'$ (and w.l.o.g. $x = a, y = b$) such that
$\sigma(a) \neq \sigma(b) = \sigma(b')$, $(a,b) \in E(\vec{G})$, and $(a,b') \notin E(\vec{G})$, and

(b) $xy|z = bb'|a$ (and w.l.o.g. $x = b, y = b'$) such that
$\sigma(a) \neq \sigma(b) = \sigma(b')$, $b \neq b'$, $(a,b), (a,b') \in E(\vec{G})$.

In Case (a), we have $a,b \in L(T(v))$, $b' = z \in L(T(v''))$ and $(a,b) \in E(\vec{G})$. Assume, for contradiction, that there is a vertex b'' of color $\sigma(b'') = \sigma(b)$ in $L(T(v'))$. In this case, $\mathrm{lca}_T(a,b) \preceq_T v \prec_T \rho = \mathrm{lca}_T(a,b'')$ would imply that $(a,b'') \notin E(\vec{G})$. Hence, we obtain the informative triple $ab|b'' \in \mathcal{R}(\vec{G},\sigma) \subseteq \mathcal{R}^{\mathrm{B}}$ with $b'' \in L(T(v')) \subset L'$. By assumption, such a triple does not exist and thus we must have $\sigma(b) \notin \sigma(L(T(v')))$. Hence, every leaf $c \in L(T(v')) \neq \emptyset$ satisfies $\sigma(c) \neq \sigma(b) = \sigma(b')$. Since the color

$\sigma(b)$ is not present in $T(v')$ and $\operatorname{lca}_T(c,b) = \operatorname{lca}_T(c,b') = \rho$, we can conclude that $(c,b),(c,b') \in E(\vec{G})$. By Eq. (7), $bb'|c \in \mathcal{R}^{\mathrm{B}}$ and thus, bb' is an edge in H. However, as argued above, b and b' lie in distinct connected components C and C'' of H; a contradiction.

In Case (b), we have $b, b' \in L(T(v))$, $a = z \in L(T(v''))$ and $(a,b),(a,b') \in E(\vec{G})$. The latter implies that the color $\sigma(b)$ is not present in the subtree $T(v'')$.

Now assume, for contradiction, that $\sigma(b)$ is not present in $T(v')$ either. Then, $(c,b),(c,b') \in E(\vec{G})$ for any $c \in L(T(v')) \neq \emptyset$, thus $bb'|c \in \mathcal{R}^{\mathrm{B}}$; a contradiction. Hence, there exists a vertex $b'' \in L(T(v'))$ with $\sigma(b'') = \sigma(b)$. Similarly, since $\sigma(b) \notin \sigma(L(T(v'')))$, we can conclude that $(a,b),(a,b'') \in E(\vec{G})$ and thus $bb''|a \in \mathcal{R}^{\mathrm{B}}$. This implies that bb'' is an edge in H. However, b and b'' lie in distinct connected components C and C' of H; a contradiction.

In summary, we conclude that for every edge xy in $H[L']$, there is a triple $xy|z$ with $\{x,y,z\} \subseteq L'$, and hence $xy \in E(H')$. Together with $V(H[L']) = V(H') = L'$ and $E(H') \subseteq E(H[L'])$, this implies $H' = H[L']$. ☐

Lemma 4.23. *The BRT (T,σ) of a binary-explainable BMG $(\vec{G},\sigma)$ satisfies $r(T) = \operatorname{cl}(\mathcal{R}^B(\vec{G},\sigma))$.*

Proof. First note that since $(\vec{G},\sigma)$ is binary-explainable, Thm. 4.4 ensures the consistency of $\mathcal{R}^{\mathrm{B}} := \mathcal{R}^{\mathrm{B}}(\vec{G},\sigma)$, and hence, the existence of the BRT (T,σ) and $\operatorname{cl}(\mathcal{R}^{\mathrm{B}})$. We proceed by induction on $L := V(\vec{G})$. The statement trivially holds for $|L| \in \{1,2\}$, since in this case, we clearly have $r(T) = \operatorname{cl}(\mathcal{R}^{\mathrm{B}}) = \emptyset$. Moreover, we can assume w.l.o.g. that $L = L_{\mathcal{R}^{\mathrm{B}}} := \bigcup_{t \in \mathcal{R}^{\mathrm{B}}} L(t)$ since otherwise Lemma 4.21 implies $\mathcal{R}^{\mathrm{B}} = \emptyset$. In this case, (T,σ) is the star tree on L, and again $r(T) = \operatorname{cl}(\mathcal{R}^{\mathrm{B}}) = \emptyset$.

For $|L| > 2$ and $L = L_{\mathcal{R}^{\mathrm{B}}}$ we assume that the statement is true for every binary-explainable BMG with less than $|L|$ vertices. We write $L_v := L(T(v))$ for the set of leaves in the subtree of (T,σ) rooted at v.

By construction of the BRT (T,σ) from $\mathcal{R}^{\mathrm{B}}$, there is a one-to-one correspondence between the connected components of the Aho graph $[\mathcal{R}^{\mathrm{B}}, L]$ and the children v of the root ρ of T. For each such vertex $v \in \mathsf{child}_T(\rho)$, the digraph $\vec{G}(T(v),\sigma_{|L_v})$ is a BMG and, by Lemma 4.9, $\vec{G}(T(v),\sigma_{|L_v}) = (\vec{G}[L_v],\sigma_{|L_v})$. Moreover, we have $\mathcal{R}^{\mathrm{B}}{}_{|L_v} = \mathcal{R}^{\mathrm{B}}(\vec{G}[L_v],\sigma_{|L_v})$ by Obs. 4.6. By the recursive construction of (T,σ) via `BUILD`, we therefore conclude that $(T(v),\sigma_{|L_v})$ is the BRT for the BMG $(\vec{G}[L_v],\sigma_{|L_v})$. By induction hypothesis, we can therefore conclude $r(T(v)) = \operatorname{cl}(\mathcal{R}^{\mathrm{B}}(\vec{G}[L_v],\sigma_{|L_v}))$.

Let $ab|c \in r(T)$ and suppose first $\operatorname{lca}_T(\{a,b,c\}) \preceq_T v \prec_T \rho$ for some $v \in \mathsf{child}_T(\rho)$. In this case, we have $ab|c \in r(T(v)) = \operatorname{cl}(\mathcal{R}^{\mathrm{B}}(\vec{G}[L_v],\sigma_{|L_v}))$. Together with $\mathcal{R}^{\mathrm{B}}(\vec{G}[L_v],\sigma_{|L_v}) = \mathcal{R}^{\mathrm{B}}{}_{|L_v} \subseteq \mathcal{R}^{\mathrm{B}}$ and monotonicity of the closure it follows $ab|c \in \operatorname{cl}(\mathcal{R}^{\mathrm{B}})$.

It remains to show that, for each triple $ab|c \in r(T)$ with $\operatorname{lca}_T(\{a,b,c\}) = \rho$, it also holds $ab|c \in \operatorname{cl}(\mathcal{R}^{\mathrm{B}})$. In this case, we have $a,b \in L_v$ and $c \in L_{v'}$ for two distinct children v and v' of the root ρ. As argued above, L_v and $L_{v'}$ correspond to two distinct connected components C_v and $C_{v'}$ of $[\mathcal{R}^{\mathrm{B}}, L]$. Consider the set $L' := L_v \cup L_{v'} = V(C_v) \cup V(C_{v'})$. By Lemma 4.22, the Aho graph $[\mathcal{R}^{\mathrm{B}}{}_{|L'}, L']$ consists exactly of the two connected components C_v and $C_{v'}$, where C_v contains a and b, and $C_{v'}$ contains c. This and the fact that $L = L_{\mathcal{R}^{\mathrm{B}}}$ allows us to apply Prop. 2.4 and to conclude that $ab|c \in \operatorname{cl}(\mathcal{R}^{\mathrm{B}})$.

In summary, every triple in $ab|c \in r(T)$ satisfies $ab|c \in \operatorname{cl}(\mathcal{R}^{\mathrm{B}})$, thus $r(T) \subseteq \operatorname{cl}(\mathcal{R}^{\mathrm{B}})$. On the other hand, the fact that T displays $\mathcal{R}^{\mathrm{B}}$ and that $r(T)$ is closed imply $\operatorname{cl}(\mathcal{R}^{\mathrm{B}}) \subseteq \operatorname{cl}(r(T)) = r(T)$. Therefore, $\operatorname{cl}(\mathcal{R}^{\mathrm{B}}) = r(T)$. ☐

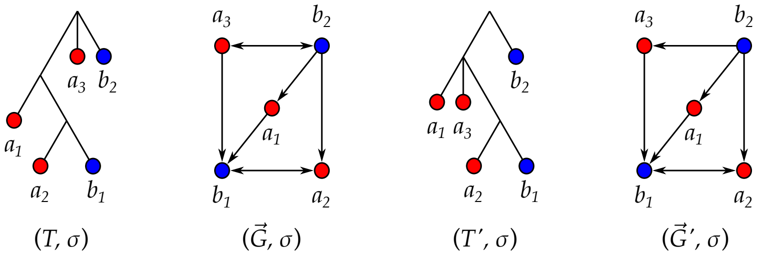


Fig. 15. A least resolved tree (T,σ) explaining the BMG $(\vec{G},\sigma)$ with informative triples $\mathcal{R} := \mathcal{R}(\vec{G},\sigma) = \{a_2b_1|a_1,\ a_2b_1|a_3,\ a_2b_1|b_2,\ a_1b_1|b_2\}$ for which $r(T) \neq \mathrm{cl}(\mathcal{R})$. The tree (T',σ) also displays $\mathcal{R}$ but $a_1a_2|a_3 \in r(T)$ and $a_1a_2|a_3 \notin r(T')$. In particular, (T',σ) explains a different BMG $(\vec{G}',\sigma)$ in which the arc (a_3, b_2) is missing.

No analog of Lemma 4.23 holds for LRTs, i.e., in general we have $\mathrm{cl}(\mathcal{R}(\vec{G},\sigma)) \neq r(T)$ for the LRT (T,σ) of a BMG $(\vec{G},\sigma)$. Fig. 15 shows a counterexample.

Recall that a set of triples $\mathcal{R}$ identifies a tree T, if every other tree (on $L(T)$) that displays $\mathcal{R}$ is a refinement of T (see Sec. 2.3.4). From Prop. 2.5 and Lemma 4.23, we immediately obtain the main result of this section:

Theorem 4.5. *Let $(\vec{G},\sigma)$ be a binary-explainable BMG with vertex set L and BRT (T,σ). Then every tree on L that displays $\mathcal{R}^B(\vec{G},\sigma)$ is a refinement of (T,σ). In particular, every binary tree that explains $(\vec{G},\sigma)$ is a refinement of (T,σ).*

Corollary 4.12. *If $(\vec{G},\sigma)$ is binary-explainable with BRT (T,σ), then a binary tree (T',σ) explains $(\vec{G},\sigma)$ if and only if it is a refinement of (T,σ).*

Assuming that evolution of a gene family only progresses by bifurcations and that the correct BMG $(\vec{G},\sigma)$ is known, Cor. 4.12 implies that the true (binary) gene tree displays the BRT of $(\vec{G},\sigma)$. Fig. 16 shows the LRT and BRT for the BMG $(\vec{G},\sigma)$. The BRT is more finely resolved than the LRT. The difference arises from the triple $a_2a_3|c_2 \in \mathcal{R}^B(\vec{G},\sigma) \setminus \mathcal{R}(\vec{G},\sigma)$. The true gene tree (Fig. 16B) is a binary refinement of the BRT (and thus also of the LRT).

4.3.4 *Simulation Results*

As argued in the previous sections, best match graphs contain valuable information on the (rooted) gene tree topology since both their LRTs and BRTs are displayed by the latter (cf. [102] and Cor. 4.12). Hence, they are of interest for the reconstruction of gene family histories. In order to illustrate the potential benefit of using the better resolved BRT instead of the LRT, we simulated realistic, but idealized, evolutionary scenarios using the library `AsymmeTree` [293], i.e., we extracted the "true" BMGs from the simulated gene trees. Hence, we do not take into account errors arising in the approximation of best matches from sequence data. In real-life applications, of course, factors such as rate variation among different branches and inaccuracies in sequence alignment need to be taken into account, see e.g. [293] and Chapter 5 for a more detailed discussion of this topic.

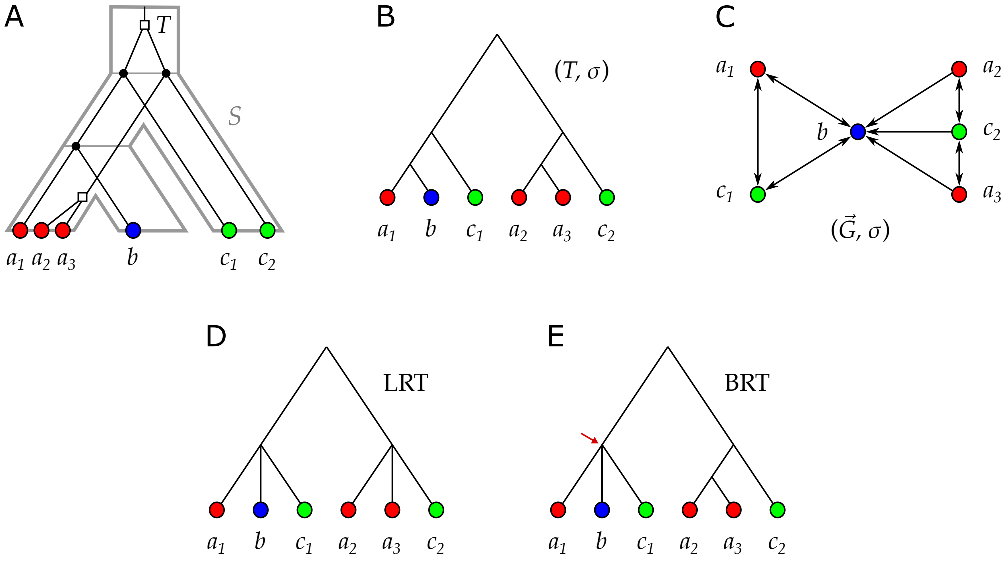


Fig. 16. (A) An evolutionary scenario consisting of a gene tree (T, σ) (whose topology is again shown in (B)) together with an embedding into a species tree S. (C) The best match graph $(\vec{G}, \sigma)$ explained by (T, σ). (D) The unique least resolved tree (LRT) explaining $(\vec{G}, \sigma)$. (E) The binary-refinable tree (BRT) is better resolved than the LRT. The remaining polytomy in the BRT (red arrow) can be resolved arbitrarily. Out of the three possibilities, one results in the original binary tree in (B).

Pairs of binary planted species and gene trees are simulated as described in Sec. 3.4. The number of species leaves is chosen at random from a range of 10 to 30, and various combinations of event rates are considered as specified in Fig. 17. To avoid trivial cases and to ensure that all colors appear in the resulting BMGs, losses are constrained in such a way that every branch (and in particular every leaf) of S has at least one surviving gene. From the observable gene tree (T, σ), the BMG and its LRT and BRT are constructed.

We consider single leaves and the full set L as trivial clades since they appear in any phylogenetic tree $T = (V, E)$ with leaf set L. We can quantify the resolution $\operatorname{res}(T)$ as the fraction of non-trivial clades of T retained in the LRT or BRT, respectively, which is the same as the fraction of inner edges that remain uncontracted. To see this, we note that T has between 0 and $|L| - 2$ edges that are not incident with leaves, with the maximum attained if and only if T is binary. Thus T has $|E| - |L|$ edges that have remained uncontracted. On the other hand, each vertex of T that is not a leaf or the root defines a non-trivial clade. Thus T contains $|V| - 1 - |L|$ non-trivial clades. Since $|E| = |V| - 1$ we have

$$\operatorname{res}(T) := \frac{|E| - |L|}{|L| - 2} = \frac{|V| - |L| - 1}{|L| - 2}. \tag{8}$$

The parameter $\operatorname{res}(T)$ is well-defined for $|L| > 2$, which is always the case in the simulated scenarios. It satisfies $\operatorname{res}(T) = 0$ for a tree consisting only of the root and leaves, and $\operatorname{res}(T) = 1$ for binary trees. Since the true gene tree (T, σ) is binary, it displays both the LRT and BRT of its BMG. Thus we have $0 \leq \operatorname{res}(\text{LRT}) \leq \operatorname{res}(\text{BRT}) \leq \operatorname{res}(T) = 1$.

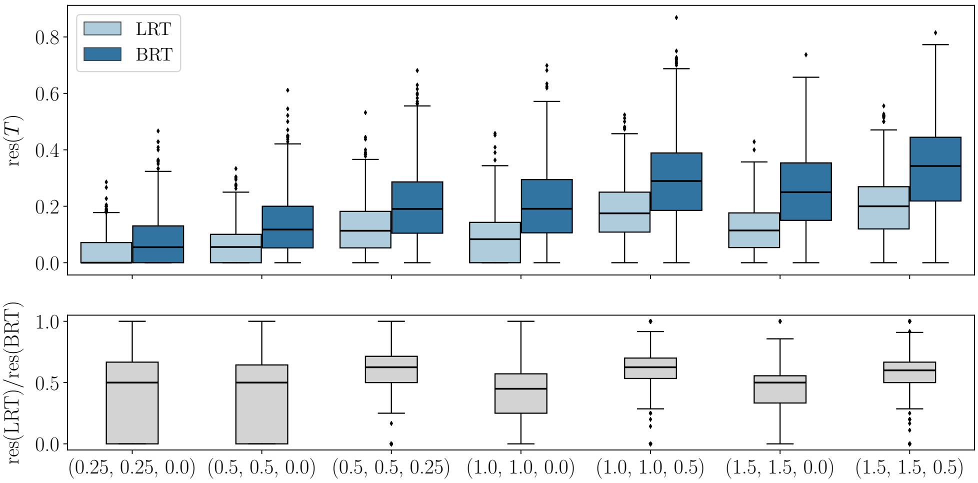


Fig. 17. Comparison of LRTs and BRTs of BMGs obtained from simulated evolutionary scenarios with 10 to 30 species and binary gene trees with different combinations of rates for gene duplications, gene loss, and horizontal transfer (indicated as triples on the horizontal axis). Top: Fraction of resolved non-trivial clades res(LRT) and res(LRT). Below: The ratio of these parameters. Distributions are computed from 1000 scenarios for each combination of rates. The box plots show the median and inter-quartile range.

The results for the simulated scenarios with different rates for duplications, losses, and horizontal transfers are summarized in Fig. 17. In general, the BRT is much better resolved than the LRT with the median values of res(BRT) exceeding res(LRT) by about a factor of two (cf. lower panel).

By Obs. 4.2, the subgraphs of BMGs induced by all vertices of two selected colors are again BMGs. In particular, a BMG can be viewed as the edge-disjoint union of these 2-colored induced subgraphs. Conceptually, this corresponds to the extraction of best hits in graph-based orthology detection by comparing two genomes at a time. The comparatively simple structure of 2-BMGs has been the key for deriving mathematical results that could then be generalized to BMGs with more colors, see e.g. [102], [137] and the complexity results for BMG modification that will be presented in Chapter 5 of this work. Moreover, this promises the potential development of practical applications for best matches that first operate on pairs of two colors and then combine the results while somehow ensuring consistency. For these reasons, 2-colored BMGs recently received some attention in the literature [102, 174, 175].

This section is organized as follows: In Sec. 4.4.1, we briefly review the previously established characterizations of 2-BMGs. In Sec. 4.4.2, we translate one of them into a set of forbidden induced subgraphs resulting in a novel characterization of 2-BMGs which appeared previously in [266]. The remainder of the section is based on [262]. We are concerned with the peculiar properties of the LRTs of 2-BMGs in Sec. 4.4.3. To this end, we introduce the concept of "support leaves" that uniquely determine the LRT. The main result of this section is Thm. 4.8, which shows that the support leaves of the root can be identified directly in the 2-BMG. In Sec. 4.4.4, we then turn Thm. 4.8 into an efficient algorithm for recognizing 2-BMGs and constructing their LRTs. Computational experiments demonstrate the performance gain in practice. In Sec. 4.4.5, we extend the algorithmic approach to binary-explainable 2-BMGs, a subclass of 2-BMGs featuring the hourglass as an additional forbidden induced subgraph.

4.4.1 *Recent Work on 2-BMGs*

Two-colored BMGs have first been characterized in [102] in two different ways. One way of recognizing a 2-colored digraph $(\vec{G} = (V,E),\sigma)$ as a 2-BMG is the extraction of its informative triple set $\mathcal{R}(\vec{G},\sigma)$, and the comparison of the BMG $\vec{G}(T^*,\sigma)$ of the Aho tree $T^* := \mathrm{Aho}(\mathcal{R}(\vec{G},\sigma),V)$ with the original digraph $(\vec{G},\sigma)$ (cf. [102, Thm. 6] and Thm. 4.2 in this work). The second characterization presented in [102] is formulated in terms of the so-called *thinness relation* which is defined on the vertices V:

Definition 4.13. *Two vertices $x,y \in V$ are in relation $\overset{\bullet}{\sim}$ if $N(x) = N(y)$ and $N^-(x) = N^-(y)$.*

A digraph is *thin* if no two vertices have the same neighborhood. Clearly, the thinness relation $\overset{\bullet}{\sim}$ is an equivalence relation on V. For each $\overset{\bullet}{\sim}$ class α, we have $N(\alpha) = N(x)$ and $N^-(\alpha) = N^-(x)$ for all $x \in \alpha$. The following characterization of 2-BMGs makes use of the structure of the trees by which they are explained. These properties can be expressed in terms of properties of the vertex neighborhoods in the 2-BMGs.

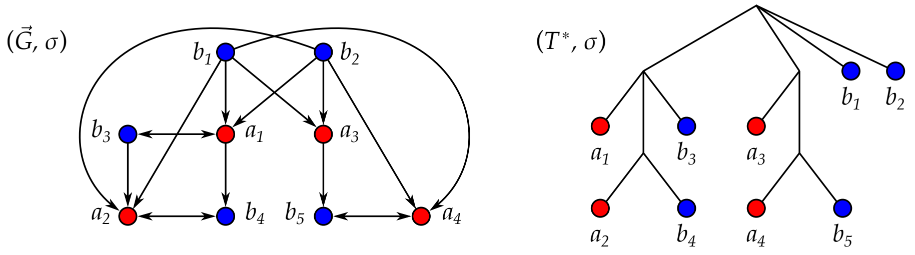


Fig. 18. Example for a 2-BMG $(\vec{G},\sigma)$ and its explaining least resolved tree (T^*,σ).

Theorem 4.6. [102, Thm. 3 and 4] *Let* $(\vec{G},\sigma)$ *be a connected properly 2-colored digraph. Then,* $(\vec{G},\sigma)$ *is a BMG if and only if for any two* $\dot{\sim}$ *classes* α *and* β *of* $\vec{G}$ *holds*

(N0) $N(\alpha) \neq \emptyset$,

(N1) $\alpha \cap N(\beta) = \beta \cap N(\alpha) = \emptyset$ *implies* $N(\alpha) \cap N(N(\beta)) = N(\beta) \cap N(N(\alpha)) = \emptyset$,

(N2) $N(N(N(\alpha))) \subseteq N(\alpha)$, *and*

(N3) $\alpha \cap N(N(\beta)) = \beta \cap N(N(\alpha)) = \emptyset$ *and* $N(\alpha) \cap N(\beta) \neq \emptyset$ *implies* $N^-(\alpha) = N^-(\beta)$ *and* $N(\alpha) \subseteq N(\beta)$ *or* $N(\beta) \subseteq N(\alpha)$.

We note that [102] tacitly assumed (N0), i.e., that $(\vec{G},\sigma)$ is *sink-free*. While all conditions can be tested in polynomial time, some of them are far from being intuitive. However, the four properties relate 2-BMGs to previously studied classes of digraphs, see also [174] for a detailed discussion.

Property (N0) states that $\vec{G}$ is *sink-free*, i.e., every vertex has at least one out-neighbor. Note that BMGs will in general have sources, i.e., $N^-(x)$ may be empty. Sink-free graphs have appeared in particular in the context of graph semigroups [5] and graph orientation problems [56]. Since they are sink-free and properly colored, 2-BMGs can alternatively be defined as the class of bipartite digraphs $\vec{G}$ that satisfy Properties (N0)–(N3), i.e., without an *a priori* assignment of a coloring σ. This definition is almost equivalent to Def. 4.2 (restricted to 2-colored digraphs) since it induces a coloring that is unique only up to relabeling of the colors independently on each (weakly) connected component of $\vec{G}$.

Property (N2) has also been described earlier in graph theory literature. It can be rephrased as follows: For any four vertices u_1, u_2, v_1, v_2 with $(u_1, v_1), (v_1, u_2), (u_2, v_2) \in E$ we have $(u_1, v_2) \in E$, i.e., $\vec{G}$ is *bi-transitive*. Bi-transitive graphs were introduced in [62] in the context of oriented bipartite graphs and investigated in more detail in [174, 175].

Based on Thm. 4.6, Geiß et al. [102] derived an $O(|V|^3)$-time algorithm for 2-BMG recognition that first checks Properties (N0)–(N3), and, if they are satisfied, constructs the hierarchy on V that corresponds to the unique least resolved tree of 2-BMG $(\vec{G},\sigma)$. Fig. 18 shows an example for a 2-BMG together with its LRT.

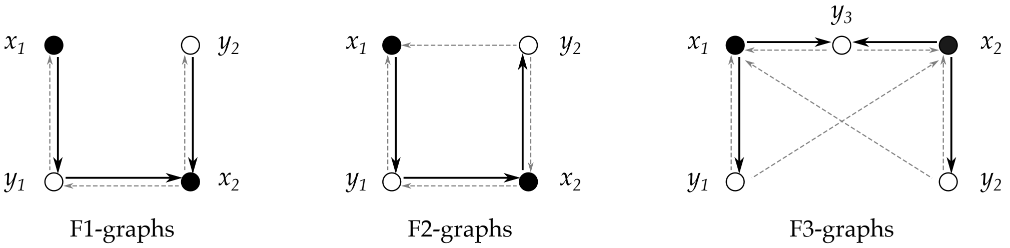


Fig. 19. Templates of the three families of forbidden induced subgraphs in BMGs. Black arcs must exist, non-arcs must not exist and dashed gray arcs may or may not be present.

4.4.2 *Forbidden Induced Subgraphs of 2-BMGs*

In this section, we derive a new characterization of 2-colored BMGs in terms of forbidden induced subgraphs. Our starting point is the observation that certain constellations of arcs on four or five vertices cannot occur.

Definition 4.14 (F1-, F2-, and F3-graphs)**.**

(F1) *A properly 2-colored digraph on four distinct vertices* $V = \{x_1, x_2, y_1, y_2\}$ *with coloring* $\sigma(x_1) = \sigma(x_2) \neq \sigma(y_1) = \sigma(y_2)$ *is an* F1-graph *if* $(x_1, y_1), (y_2, x_2), (y_1, x_2) \in E$ *and* $(x_1, y_2), (y_2, x_1) \notin E$.

(F2) *A properly 2-colored digraph on four distinct vertices* $V = \{x_1, x_2, y_1, y_2\}$ *with coloring* $\sigma(x_1) = \sigma(x_2) \neq \sigma(y_1) = \sigma(y_2)$ *is an* F2-graph *if* $(x_1, y_1), (y_1, x_2), (x_2, y_2) \in E$ *and* $(x_1, y_2) \notin E$.

(F3) *A properly 2-colored digraph on five distinct vertices* $V = \{x_1, x_2, y_1, y_2, y_3\}$ *with coloring* $\sigma(x_1) = \sigma(x_2) \neq \sigma(y_1) = \sigma(y_2) = \sigma(y_3)$ *is an* F3-graph *if* $(x_1, y_1), (x_2, y_2), (x_1, y_3), (x_2, y_3) \in E$ *and* $(x_1, y_2), (x_2, y_1) \notin E$.

The "templates" for F1-, F2-, and F3-graphs are shown in Fig. 19. They define 8, 16, and 64 digraphs by specifying the presence or absence of the 3, 4, and 6 optional (dashed) arcs, respectively, see Figs. A.10 and A.11 in [266] for a full enumeration. The F1- and F2-graphs fall into a total of 16 isomorphism classes, four of which are both F1- and F2-graphs. All but one of the F3-graphs contain an F1- or an F2-graph as induced subgraph. The exception is the "template" of the F3-graphs without optional arcs. The 17 non-redundant forbidden subgraphs are collected in Fig. 20. We shall see below that they are sufficient to characterize 2-BMGs among the sink-free digraphs.

Lemma 4.24. *If* $(\vec{G}, \sigma)$ *is a BMG, then it contains no induced F1-, F2-, or F3-graph.*

Proof. Let (T, σ) be a tree that explains $(\vec{G}, \sigma)$.

First, assume that $(\vec{G}, \sigma)$ contains an induced F1-graph, i.e., there are four vertices x_1, x_2, y_1, y_2 satisfying (F1), and let $u := \operatorname{lca}_T(x_1, y_2)$. Then, $(x_1, y_1), (y_2, x_2) \in E$, $(x_1, y_2), (y_2, x_1) \notin E$ and Lemma 4.1 imply that T must display the informative triples $x_1 y_1 | y_2$ and $y_2 x_2 | x_1$. Hence, u must have two distinct children v_1 and v_2 such that $x_1, y_1 \prec_T v_1$ and $x_2, y_2 \prec_T v_2$ by Lemma 4.5. Therefore, $\operatorname{lca}_T(x_1, y_1) \preceq_T v_1 \prec_T u = \operatorname{lca}_T(x_2, y_1)$ and $\sigma(x_1) = \sigma(x_2)$ imply that $(y_1, x_2) \notin E(\vec{G})$; a contradiction.

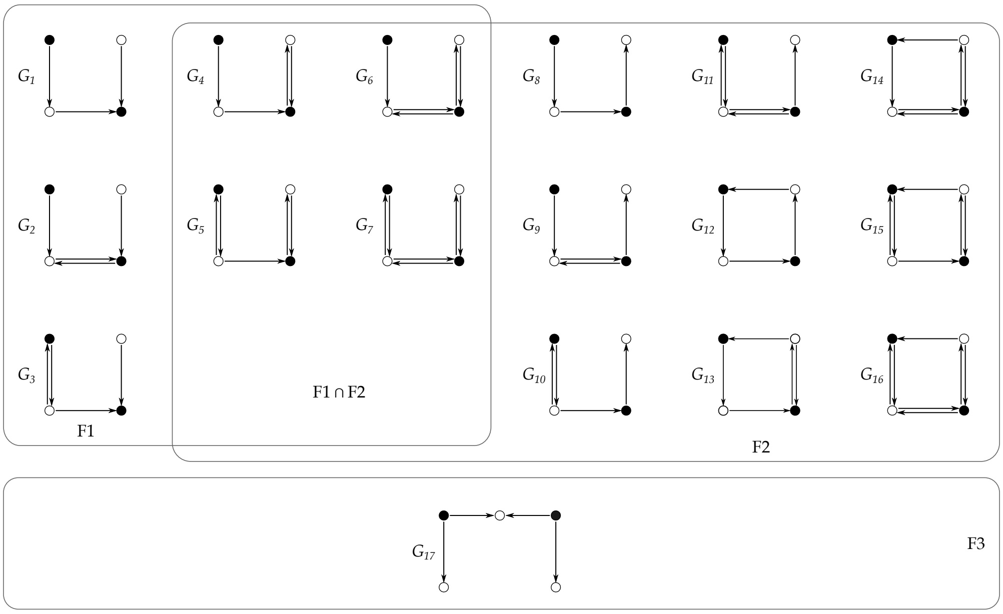


Fig. 20. Forbidden induced subgraphs in BMGs. All F3-graphs with at least one optional arc have an induced F1- or F2-graph and thus are redundant.

Next, assume that $(\vec{G},\sigma)$ contains an induced F2-graph, i.e., there are four vertices x_1, x_2, y_1, y_2 satisfying (F2). Then $(x_1, y_1) \in E$, $(x_1, y_2) \notin E$ and Lemma 4.1 imply that T displays the informative triple $x_1 y_1 | y_2$ and thus $\operatorname{lca}_T(x_1, y_1) \prec_T \operatorname{lca}_T(x_1, y_2)$. Since $(y_1, x_2) \in E$ and $\sigma(x_1) = \sigma(x_2)$, we conclude that $\operatorname{lca}_T(x_2, y_1) \preceq_T \operatorname{lca}_T(x_1, y_1) \prec_T \operatorname{lca}_T(x_1, y_2)$ and therefore also $\operatorname{lca}_T(x_2, y_1) \prec_T \operatorname{lca}_T(x_2, y_2) = \operatorname{lca}_T(x_1, y_2)$. Together with $\sigma(y_1) = \sigma(y_2)$, the latter contradicts $(x_2, y_2) \in E$.

Finally, assume that $(\vec{G},\sigma)$ contains an induced F3-graph, i.e., there are five vertices x_1, x_2, y_1, y_2, y_3 satisfying (F3). By Lemmas 4.1 and 4.5, $(x_1, y_1) \in E$ and $(x_1, y_2) \notin E$ implies that T displays the triple $x_1 y_1 | y_2$, and $(x_2, y_2) \in E$ together with $(x_2, y_1) \notin E$ implies that T displays the triple $x_2 y_2 | y_1$. Furthermore, $\operatorname{lca}_T(x_1, x_2)$ has distinct children v_1 and v_2 such that $x_1, y_1 \prec_T v_1$ and $x_2, y_2 \prec_T v_1$. Now since $\sigma(y_1) = \sigma(y_2) = \sigma(y_3)$, the two arcs (x_1, y_3) and (x_2, y_3) imply that $\operatorname{lca}_T(x_1, y_3) \preceq_T \operatorname{lca}_T(x_1, y_1) \preceq_T v_1$ and $\operatorname{lca}_T(x_2, y_3) \preceq_T \operatorname{lca}_T(x_2, y_2) \preceq_T v_2$, respectively. Since v_1 and v_2 are incomparable w.r.t. $\preceq_T$, this is a contradiction. $\square$

Lemma 4.25. *Let $(\vec{G},\sigma)$ be a properly 2-colored digraph. Then $(\vec{G},\sigma)$ satisfies* (N1) *if it does not contain an induced F1-graph, it satisfies* (N2) *if it does not contain an induced F2-graph, and it satisfies* (N3) *if is contains neither an induced F1-graph nor an induced F3-graph.*

Proof. We employ contraposition and thus show that $(\vec{G} = (V, E), \sigma)$ contains a forbidden subgraph whenever (N1), (N2) or (N3) are violated.

Assume that (N1) is not satisfied. Thus, there are two $\mathrel{\dot{\sim}}$-classes α and β with $\alpha \cap N(\beta) = \beta \cap N(\alpha) = \emptyset$ for which $N(\alpha) \cap N(N(\beta)) \neq \emptyset$ or $N(\beta) \cap N(N(\alpha)) \neq \emptyset$. We can w.l.o.g. assume that $N(\beta) \cap N(N(\alpha)) \neq \emptyset$. Note that $\alpha \cap N(\beta) = \emptyset$ implies that $(y, x) \notin E$ for all $x \in \alpha, y \in \beta$. Likewise $(x, y) \notin E$ for all $x \in \alpha, y \in \beta$, since $\beta \cap N(\alpha) = \emptyset$. Let $x_1 \in \alpha$, $y_2 \in \beta$ and $x_2 \in N(\beta) \cap N(N(\alpha)) \neq \emptyset$. It must hold $(x_1, y_2), (y_2, x_1) \notin E$ by the arguments above. Since $x_2 \in N(\beta)$, we have $(y_2, x_2) \in E$. Moreover, $\sigma(x_1) = \sigma(x_2) \neq \sigma(y_2)$, since $(\vec{G},\sigma)$ is properly colored. Clearly, $x_2 \in N(N(\alpha))$ implies that $N(\alpha) \neq \emptyset$. Now, let $y_1 \in N(\alpha)$ be a vertex such that

$(y_1,x_2)\in E$, which must exist as a consequence of $x_2\in N(N(\alpha))$. We have (x_1,y_1) since $y_1\in N(\alpha)$ and thus $\sigma(y_1)=\sigma(y_2)\neq\sigma(x_1)=\sigma(x_2)$. Finally, $(y_1,x_2)\in E$ immediately implies that $y_1\neq y_2$. In summary, $(x_1,y_1),(y_1,x_2),(y_2,x_2)\in E$ and $(x_1,y_2),(y_2,x_1)\notin E$, and thus $(\vec{G},\sigma)$ contains an induced F1-graph.

Now assume that (N2) is not satisfied and thus, $N(N(N(\alpha)))\not\subseteq N(\alpha)$ for some $\overset{\bullet}{\sim}$-class α. Note, the latter implies that $N(N(N(\alpha)))\neq\emptyset$. Hence, there is a vertex $y_2\in N(N(N(\alpha)))$ such that $y_2\notin N(\alpha)$. Thus, there is a vertex $x_1\in\alpha$ such that $(x_1,y_2)\notin E$. By the definition of neighborhoods and since $y_2\in N(N(N(\alpha)))$, we find vertices $y_1\in N(\alpha)$ and $x_2\in N(N(\alpha))$ such that $(x_1,y_1),(y_1,x_2),(x_2,y_2)$. Since $(\vec{G},\sigma)$ is properly colored, we must have $\sigma(x_1)=\sigma(x_2)\neq\sigma(y_1)=\sigma(y_2)$. Moreover, $(x_1,y_2)\notin E$ together with $(x_2,y_2)\in E$ and $(x_1,y_1)\in E$ implies $x_1\neq x_2$ and $y_1\neq y_2$, respectively. We conclude that the subgraph induced by x_1,x_2,y_1,y_2 contains an induced F2-graph.

Finally, assume that (N3) is not satisfied. Hence, there are two $\overset{\bullet}{\sim}$-classes α and β with $\alpha\cap N(N(\beta))=\beta\cap N(N(\alpha))=\emptyset$ and $N(\alpha)\cap N(\beta)\neq\emptyset$, but (i) $N^-(\alpha)\neq N^-(\beta)$, or (ii) neither $N(\alpha)\subseteq N(\beta)$ nor $N(\beta)\subseteq N(\alpha)$. Note, $N(\alpha)\cap N(\beta)\neq\emptyset$ implies that there a vertices $x_1\in\alpha$ and $x_2\in\beta$ with $\sigma(x_1)=\sigma(x_2)$ since $(\vec{G},\sigma)$ is properly 2-colored. In particular, there must be a vertex y_3 with $(x_1,y_3),(x_2,y_3)\in E$ and thus $\sigma(x_1)=\sigma(x_2)\neq\sigma(y_3)$.

Now consider Case (i) and suppose that $N^-(\alpha)\neq N^-(\beta)$. Thus we can assume w.l.o.g. that there is a y^* with $(y^*,x_2)\in E$ but $(y^*,x_1)\notin E$. Note, $(x_1,y^*)\notin E$, since otherwise $(x_1,y^*),(y^*,x_2)\in E$ would contradict $\beta\cap N(N(\alpha))=\emptyset$. Thus, $y^*\neq y_3$ since $(x_1,y^*)\notin E$ and $(x_1,y_3)\in E$. Furthermore, $\sigma(y^*)=\sigma(y_3)\neq\sigma(x_1)=\sigma(x_2)$, since $(\vec{G},\sigma)$ is properly 2-colored. In summary, $(y^*,x_2),(x_1,y_3),(x_2,y_3)\in E$ and $(y^*,x_1),(x_1,y^*)\notin E$ which implies that $(\vec{G},\sigma)$ contains an induced F1-graph.

Now consider Case (ii) and assume that it holds neither $N(\alpha)\subseteq N(\beta)$ nor $N(\beta)\subseteq N(\alpha)$. Clearly, the latter implies $N(\alpha)\neq\emptyset$ and $N(\beta)\neq\emptyset$. The latter two arguments imply that there must be two distinct vertices $y_1\in N(\alpha)\setminus N(\beta)$ and $y_2\in N(\beta)\setminus N(\alpha)$ and, therefore, $(x_1,y_1),(x_2,y_2)\in E$ and $(x_1,y_2),(x_2,y_1)\notin E$. It follows that $y_1\neq y_3$ and $y_2\neq y_3$ and $\sigma(y_1)=\sigma(y_2)=\sigma(y_3)\neq\sigma(x_1)=\sigma(x_2)$. This and $(x_1,y_1),(x_2,y_2),(x_1,y_3),(x_2,y_3)\in E$ together with $(x_1,y_2),(x_2,y_1)\notin E$ implies that $(\vec{G},\sigma)$ contains an induced F3-graph. ☐

Hence, the forbidden induced F1-, F2-, and F3-subgraphs characterize exactly the class of bipartite digraphs satisfying the Properties (N1), (N2), and (N3) mentioned above. Based on the latter findings we obtain here a new characterization of 2-colored BMGs that is not restricted to connected digraphs.

Theorem 4.7. *A properly 2-colored digraph is a BMG if and only if it is sink-free and does not contain an induced F1-, F2-, or F3-graph.*

Proof. Suppose that $(\vec{G},\sigma)$ is 2-colored BMG and $\mathfrak{C}$ be the set of its connected components. By Lemma 4.24, $(\vec{G},\sigma)$ does not contain an induced F1-, F2- or F3-graph. Moreover, by Lemma 4.2, $(\vec{G}[C],\sigma_{|C})$ must be a 2-colored BMG for all $C\in\mathfrak{C}$. Hence, we can apply Thm. 4.6 to conclude that each $(\vec{G}[C],\sigma_{|C})$ satisfies (N0)-(N3). Since every $x\in V$ is contained in some $\overset{\bullet}{\sim}$-class, (N0) is equivalent to $N(x)\neq\emptyset$, i.e., $(\vec{G},\sigma)$ is sink-free.

Now suppose that $(\vec{G},\sigma)$ is properly 2-colored and sink-free, and that it does not contain an induced F1-, F2- and F3-graph. By Lemma 4.25, $(\vec{G},\sigma)$ satisfies (N1)-(N3). Thus, in particular, each connected component of $(\vec{G},\sigma)$ is sink-free and satisfies and (N1)-(N3). Note, $N(x)\neq\emptyset$ implies that the connected components of $(\vec{G},\sigma)$ contain at least one arc and, by assumption, they are properly 2-colored. Moreover, this implies that (N0) is satisfied for every connected component of $(\vec{G},\sigma)$. Hence, Thm. 4.6 implies that every connected component of $(\vec{G},\sigma)$ is a 2-colored BMG. By Lemma 4.2, $(\vec{G},\sigma)$ is also a 2-colored BMG. ☐

4.4.3 *Support Leaves*

In this section, we introduce "support leaves" as a means to recursively construct the LRT of a 2-BMG. The main result of this section shows that these leaves can be inferred directly from the BMG without any further knowledge of the corresponding LRT. In this and the following sections, we will make extensive use of restrictions of the coloring σ on $V(\vec{G})$ or $L(T)$ to subsets of vertices or leaves. Since in situations like $(\vec{G}_i,\sigma_{|V(\vec{G}_i)})$ the set to which σ is restricted is clear, we will write $\sigma_{|.}$ to keep the notation less cluttered.

We start with an interesting property of LRTs of BMGs with an arbitrary number of colors which we will use repeatedly.

Lemma 4.26. *Let (T,σ) be the least resolved tree of some BMG $(\vec{G},\sigma)$. Then every vertex $v \prec_T \rho_T$ with $|\sigma(L(T(v)))| = 1$ is a leaf.*

Proof. Let $v \prec_T \rho_T$ with $|\sigma(L(T(v)))| = 1$ and assume, for contradiction, that v is not a leaf. Hence, $|L(T(v))| > 1$. By Lemma 4.9 $\vec{G}(T(v),\sigma_{|.})$ is a BMG and, therefore, properly colored. But then $\vec{G}(T(v),\sigma_{|.})$ is disconnected; a contradiction to Lemma 4.10. ☐

As a consequence, we find

Corollary 4.13. *Let (T,σ) be the least resolved tree of some BMG $(\vec{G},\sigma)$. Then any vertex $v \in V(T)$ with $v \prec_T \rho_T$ is an inner vertex if and only if $|\sigma(L(T(v)))| > 1$.*

Proof. If $|\sigma(L(T(v)))| = 1$, Lemma 4.26 implies that v is a leaf. Otherwise, if $|\sigma(L(T(v)))| > 1$, $T(v)$ clearly must contain at least two leaves and thus v cannot be a leaf. ☐

The following two results link the out-neighborhood of a vertex in a 2-colored BMG to its position in the least resolved tree, and are needed for later reference.

Lemma 4.27. *Let $(\vec{G},\sigma)$ be a 2-colored BMG, (T,σ) its LRT and $x,y \in L(T) = V(\vec{G})$. Then $(x,y) \in E(\vec{G})$ if and only if $\sigma(x) \neq \sigma(y)$ and $y \in L(T(\mathrm{par}_T(x)))$.*

Proof. First note that, since $(\vec{G},\sigma)$ is 2-colored, (T,σ) has at least two leaves and $u := \mathrm{par}_T(x)$ is always defined. First, assume $\sigma(x) \neq \sigma(y)$, and thus $x \neq y$, and let $y \in L(T(u))$. Since x is a child of u, we have $\mathrm{lca}_T(x,y) = u$. Moreover, since u is the parent of x, there is no vertex y' of color $\sigma(y)$ such that $\mathrm{lca}_T(x,y') \prec_T \mathrm{lca}_T(x,y) = u$. Hence, y is a best match of x, i.e., $(x,y) \in E(\vec{G})$.

Now suppose, for contraposition, that $\sigma(x) = \sigma(y)$ or $y \notin L(T(u))$. If $\sigma(x) = \sigma(y)$, then, by definition, $(x,y) \notin E(\vec{G})$. If $y \notin L(T(u))$, then $u \prec_T \rho_T$. Hence, we can apply Cor. 4.13 to the inner vertex u to conclude that $|\sigma(L(T(u)))| > 1$, i.e., the subtree $L(T(u))$ contains both colors. Thus, we can find a vertex y' of color $\sigma(y)$ such that $\mathrm{lca}_T(x,y') \preceq_T u \prec_T \mathrm{lca}_T(x,y)$ which implies that $(x,y) \notin E(\vec{G})$. ☐

As an immediate consequence, we find

Corollary 4.14. *Let $(\vec{G},\sigma)$ be a 2-colored BMG, (T,σ) its LRT and $x,y \in V(\vec{G}) = L(T)$. Then $(x,y),(y,x) \in E(\vec{G})$ if and only if $\sigma(x) \neq \sigma(y)$ and $\mathsf{par}_T(x) = \mathsf{par}_T(y)$.*

We continue with a technical result similar to Cor. 3 in [102]; here we use a much simpler, more convenient notation.

Lemma 4.28. *Let (T,σ) be the least resolved tree of a 2-colored BMG $(\vec{G},\sigma)$. Then, for every vertex $u \in V^0(T) \setminus \{\rho_T\}$, it holds $\mathsf{child}_T(u) \cap L(T) \neq \emptyset$. If $(\vec{G},\sigma)$ is connected, then $\mathsf{child}_T(u) \cap L(T) \neq \emptyset$ holds for every $u \in V^0(T)$.*

Proof. Suppose first that $(\vec{G},\sigma)$ is disconnected and let $u \in V^0(T) \setminus \{\rho_T\}$. Since (T,σ) is least resolved, Lemma 4.10 implies that $\vec{G}(T(u),\sigma_{|.})$ is connected for every $u \in V(T)$ with $u \prec_T \rho_T$. Hence, we can apply Prop. 4.1 to $\vec{G}(T(u),\sigma_{|.})$ and conclude that there is a child $v \in \mathsf{child}_{T(u)}(u)$ such that $\sigma(L(T(v))) \neq \sigma(L(T(u)))$, hence in particular $\sigma(L(T(v))) \subsetneq \sigma(L(T(u)))$. Since (T,σ) is 2-colored, the latter immediately implies $|\sigma(L(T(v)))| = 1$ and, by Cor. 4.13, v is a leaf. Thus every $u \in V^0(T) \setminus \{\rho_T\}$ has a leaf v among its children, i.e. $\mathsf{child}_T(u) \cap L(T) \neq \emptyset$. If in addition $(\vec{G},\sigma)$ is connected, we can apply the same argumentation to $u = \rho_T$ and conclude that a leaf v is attached to ρ_T. ☐

Lemma 4.28 states that, in the least resolved tree of a connected 2-colored BMG, every inner vertex u is adjacent to at least one leaf, and thus in a way "supported" by it.

Definition 4.15 (Support Leaves). *For a given tree T, the set $S_u := \mathsf{child}_T(u) \cap L(T)$ is the set of all* support leaves *of vertex $u \in V(T)$.*

Note that Lemma 4.28 is in general not true for ℓ-BMGs with $\ell \geq 3$, as exemplified by the (least resolved) tree $((a,b),(c,a'))$ with three distinct leaf colors $\sigma(a) = \sigma(a') \neq \sigma(b) \neq \sigma(c)$.

As a simple consequence of Prop. 4.1 and Cor. 4.13, we find

Corollary 4.15. *Let (T,σ) be the least resolved tree (with root ρ) of some 2-colored BMG $\vec{G}(T,\sigma)$. Then, $\vec{G}(T,\sigma)$ is connected if and only if $S_\rho \neq \emptyset$.*

Proof. By Prop. 4.1, $\vec{G}(T,\sigma)$ is connected if and only if there exists a child v of the root ρ of T, $v \in \mathsf{child}_T(\rho)$, such that $T(v)$ does not contain all colors. Thus $|\sigma(L(T(v)))| = 1$. By Cor. 4.13, we have $|\sigma(L(T(v)))| = 1$ if and only if v is a leaf, i.e. $v \in S_\rho$. Hence, $\vec{G}(T,\sigma)$ is connected if and only if $S_\rho \neq \emptyset$. ☐

Lemma 4.29. *Let (T,σ) be the least resolved tree of a 2-BMG $(\vec{G},\sigma)$, and S_ρ the set of support leaves of the root ρ. Then the connected components of $(\vec{G} - S_\rho, \sigma_{|.})$ are exactly the BMGs $\vec{G}(T(v),\sigma_{|.})$ with $v \in \mathsf{child}_T(\rho) \setminus S_\rho$.*

Proof. Let $v \in \mathsf{child}_T(\rho) \cap V^0(T) = \mathsf{child}_T(\rho) \setminus S_\rho$ and consider the BMG $\vec{G}(T(v),\sigma_{|.})$. By Lemma 4.10 and Lemma 4.9, $\vec{G}(T(v),\sigma_{|.})$ is connected and we have $\vec{G}(T(v),\sigma_{|.}) = (\vec{G}[L(T(v))],\sigma_{|.})$. Moreover, it holds $((\vec{G} - S_\rho)[L(T(v))],\sigma_{|.}) = (\vec{G}[L(T(v))],\sigma_{|.})$ since $L(T(v)) = V(\vec{G}[L(T(v))]) = V(H[L(T(v))])$ for $H := \vec{G} - S_\rho = \vec{G}[V(\vec{G}) \setminus S_\rho]$.

If $\mathsf{child}_T(\rho) \setminus S_\rho = \{v\}$, then the statement is trivially satisfied. Therefore, suppose that $|\mathsf{child}_T(\rho) \setminus S_\rho| > 1$. Hence, it remains to show that there are no arcs between $\vec{G}(T(v),\sigma_{|.})$ and $\vec{G}(T(w),\sigma_{|.})$ for any $w \in \mathsf{child}_T(\rho) \setminus S_\rho$, $w \neq v$. Cor. 4.13 and $v \prec_T \rho$

imply that $T(v)$ contains both colors. Thus, by Obs. 4.3, there are no out-arcs to any vertex in $L(T) \setminus L(T(v))$, hence in particular there are no out-arcs (x,y) with $x \preceq_T v$, $y \preceq_T w$. By symmetry, the same holds for w, thus we can conclude that there are no arcs (y,x). From the observation that each $x \in L(T) \setminus S_\rho$ must be located below some $v \in \mathsf{child}_T(\rho) \cap V^0(T)$, it now immediately follows that $(\vec{G} - S_\rho, \sigma_{|.})$ consists exactly of these connected components as stated. □

As a consequence, we have

Corollary 4.16. *Let (T,σ) with root ρ be the LRT of a 2-BMG $(\vec{G},\sigma)$. Then each child of ρ is either one of the support leaves S_ρ of ρ or the root of the LRT for a connected component of $(\vec{G} - S_\rho, \sigma_{|.})$.*

Proof. Let (T,σ) with root ρ be the least resolved tree for $(\vec{G},\sigma)$. The support leaves S_ρ are children of ρ by definition. By Lemma 4.29, the connected components of $(\vec{G} - S_\rho, \sigma_{|.})$ are exactly the BMGs $\vec{G}(T(v), \sigma_{|.})$ with $v \in \mathsf{child}_T(\rho) \setminus S_\rho$. Moreover, by Lemma 4.9, the subtrees $T(v)$ with $v \in \mathsf{child}_T(\rho) \setminus S_\rho$ are exactly the unique LRTs for these BMGs. □

In order to use this property as a means of constructing the LRT in a recursive manner, we need to identify the support leaves of the root S_ρ directly from the 2-BMG $(\vec{G},\sigma)$ without constructing the LRT first. To this end, we consider the set of *umbrella vertices* $U(\vec{G},\sigma)$ comprising all vertices x for which $N(x)$ consists of *all* vertices of $V(\vec{G})$ that have the color distinct from $\sigma(x)$.

Definition 4.16 (Umbrella Vertices)**.** *For an arbitrary 2-colored digraph $(\vec{G},\sigma)$, the set*

$$U(\vec{G},\sigma) := \left\{ x \in V(\vec{G}) \mid y \in N(x) \textit{ if } \sigma(y) \neq \sigma(x) \textit{ and } y \in V(\vec{G}) \right\}$$

is the set umbrella vertices *of $(\vec{G},\sigma)$.*

The intuition behind this definition is that every support leaf of the root of the LRT of a 2-BMG must have all differently colored vertices as out-neighbors, i.e., they are umbrella vertices. We now define "support sets" of digraphs as particular subsets of umbrella vertices. As we shall see later, support sets are closely related to support vertices in S_ρ.

Definition 4.17 (Support Set of $(\vec{G},\sigma)$)**.** *Let $(\vec{G},\sigma)$ be a 2-colored digraph. A* support set *$S := S(\vec{G},\sigma)$ of $(\vec{G},\sigma)$ is a maximal subset $S \subseteq U(\vec{G},\sigma)$ of umbrella vertices such that $x \in S$ implies $N^-(x) \subseteq S$.*

Lemma 4.30. *Every 2-colored digraph $(\vec{G},\sigma)$ has a unique support set $S(\vec{G},\sigma)$.*

Proof. Assume, for contradiction, that $(\vec{G},\sigma)$ has (at least) two distinct support sets $S, S' \subseteq U(\vec{G},\sigma)$. Clearly neither of them can be a subset of the other, since supports sets are maximal. We have $N^-(x) \subseteq S$ for all $x \in S$ and $N^-(x') \subseteq S'$ for all $x' \in S'$, which implies that $N^-(z) \subseteq S \cup S'$ for all $z \in S \cup S'$. Together with the fact that S, S', and thus $S \cup S'$, are all subsets of $U(\vec{G},\sigma)$, this contradicts the maximality of both S and S'. □

For the construction of the support set $S := S(\vec{G},\sigma)$, we consider the following sequence of sets, defined recursively by

$$S^{(k)} := \{x \in S^{(k-1)} \mid N^{-}(x) \subseteq S^{(k-1)}\} \text{ for } k \geq 1 \text{ and } S^{(0)} = U(\vec{G},\sigma). \quad (9)$$

By construction $S^{(k+1)} \subseteq S^{(k)}$. Furthermore, there is a $k < |V(\vec{G})|$ such that $S^{(k+1)} = S^{(k)}$. Next we show that in a 2-BMG, S is obtained in a single iteration.

Lemma 4.31. *If $(\vec{G},\sigma)$ is a 2-BMG, then $S = S^{(1)}$.*

Proof. Let $(\vec{G} = (V,E),\sigma)$ be a 2-BMG and $U = U(\vec{G},\sigma)$. Assume for contradiction that $S \neq S^{(1)}$, and thus $S^{(2)} \subsetneq S^{(1)}$. We will show that this implies the existence of a forbidden F2-graph. By assumption, there is a vertex $x_2 \in S^{(1)} \setminus S^{(2)}$. Thus, there must be a vertex $y_1 \in N^{-}(x_2)$ (and thus $(y_1,x_2) \in E$) with $\sigma(y_1) \neq \sigma(x_2)$ such that $y_1 \notin S^{(1)}$. However, by definition, $y_1 \in N^{-}(x_2)$ and $x_2 \in S^{(1)}$ implies $y_1 \in U$. Now, it follows from $y_1 \in U \setminus S^{(1)}$ that there is a vertex $x_1 \in N^{-}(y_1)$ with $\sigma(x_1) = \sigma(x_2) \neq \sigma(y_1)$ such that $x_1 \notin U$. The latter together with $x_2 \in S^{(1)} \subseteq U$ implies $x_1 \neq x_2$. In particular, since $x_1 \notin U$, the vertex x_1 does not have an out-arc to every differently colored vertex, thus there must be a vertex y_2 with $\sigma(y_2) = \sigma(y_1)$ such that $(x_1,y_2) \notin E$. Since $x_1 \in N^{-}(y_1)$, we have $(x_1,y_1) \in E$ and $y_1 \neq y_2$. Finally, $x_2 \in U$ and $\sigma(y_2) = \sigma(y_1) \neq \sigma(x_2)$ implies that $(x_2,y_2) \in E$. In summary, we have four distinct vertices x_1,x_2,y_1,y_2 with $\sigma(x_1) = \sigma(x_2) \neq \sigma(y_1) = \sigma(y_2)$ and (non-)arcs $(x_1,y_1),(y_1,x_2),(x_2,y_2) \in E$ and $(x_1,y_2) \notin E$, and hence an induced F2-graph in $(\vec{G},\sigma)$. By Thm. 4.7, we can conclude that $(\vec{G},\sigma)$ is not a BMG; a contradiction. ☐

In general, $S = S^{(0)} = U(\vec{G},\sigma)$ is not satisfied. To see this, consider the BMG $(\vec{G},\sigma)$ that is explained by the triple $x_1y|x_2$ with $\sigma(x_1) = \sigma(x_2) \neq \sigma(y)$. One easily verifies that $U(\vec{G},\sigma) = \{x_1,x_2\}$ but $S = \{x_2\}$.

Theorem 4.8. *Let (T,σ) be the least resolved tree of a 2-BMG $(\vec{G},\sigma)$. Then, the set of support leaves S_ρ of the root ρ equals the support set S of $(\vec{G},\sigma)$. In particular $S \neq \emptyset$ if and only if $(\vec{G},\sigma)$ is connected.*

Proof. Let (T,σ) be the LRT of a 2-BMG $(\vec{G} = (V,E),\sigma)$. We set $U := U(\vec{G},\sigma)$ and note first that $S = S^{(1)}$ by Lemma 4.31.

First, suppose that $(\vec{G},\sigma)$ is not connected. Then it immediately follows from Prop. 4.1 that $\sigma(L(T(v))) = \sigma(L(T))$ and thus $|\sigma(L(T(v)))| > 1$ for any $v \in \mathsf{child}_T(\rho)$. The latter together with Cor. 4.13 implies that any child of ρ must be an inner vertex in T. Hence, $S_\rho = \emptyset$. On the other hand, since $(\vec{G},\sigma)$ is not connected, each of its connected components is a 2-BMG (cf. Lemma 4.2), and thus, contains both colors. Therefore, for each vertex x in $\vec{G}$, we can find a vertex y with $\sigma(x) \neq \sigma(y)$ such that $(x,y),(y,x) \notin E$, and thus $x \notin S$. Since this is true for any vertex in $\vec{G}$, we can conclude $S = \emptyset = S_\rho$.

Now, suppose that $(\vec{G},\sigma)$ is connected. By Cor. 4.15, we have $S_\rho \neq \emptyset$. We first show $S_\rho \subseteq S$. Let $x \in S_\rho$. By definition, x satisfies $\mathrm{lca}_T(x,y) = \rho$ and therefore $(x,y) \in E$ for all $y \in L(T)$ with $\sigma(y) \neq \sigma(x)$, i.e., x has an out-arc to every differently colored vertex in $\vec{G}$. By definition, we thus have $x \in U$. Now assume for contradiction that $x \notin S = S^{(1)} = \{z \in U \mid N^{-}(z) \subseteq U\}$. The latter implies that there exists a vertex $y \in N^{-}(x)$ such that $y \notin U$. In particular, $(y,x) \in E$. Since $y \notin U$, there is some vertex x' with $\sigma(x') = \sigma(x)$ such that $(y,x') \notin E$. Together this implies that $xy|x'$ is an informative triple. By Lemma 4.1, we obtain $\mathrm{lca}_T(x,y) \prec_T \mathrm{lca}_T(x,x') = \mathrm{lca}_T(x',y) \preceq_T \rho$; a contradiction to the assumption that x is a support leaf of ρ. Thus $x \in S$.

Next, we show by contraposition that $S \subseteq S_\rho$. To this end, suppose that x is not a support leaf of ρ, i.e. $x \notin S_\rho$. Hence, there is an inner vertex $v \in \mathsf{child}_T(\rho) \cap V^0(T)$ such that $x \prec_T v$. By Cor. 4.13, we conclude that $|\sigma(L(T(v)))| = 2$, i.e., the subtree $T(v)$ contains both colors. We now distinguish two cases: (i) there is a leaf $y' \in L(T) \setminus L(T(v))$ with $\sigma(y') \neq \sigma(x)$, and (ii) there is no leaf $y' \in L(T) \setminus L(T(v))$ with $\sigma(y') \neq \sigma(x)$.

Case(i): Since $T(v)$ contains both colors, there is a leaf $y \in L(T(v))$, with $y \neq y'$ and $\sigma(y) = \sigma(y') \neq \sigma(x)$. Since, by construction, we have $\mathrm{lca}_T(x,y) \preceq_T v \prec_T \rho = \mathrm{lca}_T(x,y')$, it follows $(x,y') \notin E$. Together with $\sigma(x) \neq \sigma(y')$, this immediately implies $x \notin U$. From $S^{(2)} \subseteq S^{(1)} \subseteq U$, we conclude $x \notin S^{(1)} = S$.

Case(ii): Suppose that there is no leaf $y' \in L(T) \setminus L(T(v))$ with $\sigma(y') \neq \sigma(x)$. We will continue by showing that there is a support leaf y of vertex v with $\sigma(y) \neq \sigma(x)$. Assume, for contradiction, that the latter is not the case. Since (T,σ) is least resolved, the inner edge ρv is not redundant. Hence, by Lemma 4.6, there must be an arc $(a,b) \in E$ such that $\mathrm{lca}_T(a,b) = v$ and $\sigma(b) \in \sigma(L(T) \setminus L(T(v)))$. Since there is no leaf $y' \in L(T) \setminus L(T(v))$ with $\sigma(y') \neq \sigma(x)$, we conclude that $\sigma(b) = \sigma(x)$ and $\sigma(a) \neq \sigma(x)$. Clearly, it holds $a, b \in L(T(v))$. Now consider an arbitrary $a' \in L(T(v))$ with $\sigma(a') \neq \sigma(x)$. Since, by assumption, every such a' is not a support leaf of v, there must be an inner vertex $w \in \mathsf{child}_{T(v)}(v)$ with $a' \prec_T w$. By Cor. 4.13 and since $w \prec_T v \prec_T \rho$, we conclude that $|\sigma(L(T(w)))| = 2$, i.e., the subtree $T(w)$ contains both colors. Thus there is some b' with $\sigma(b') = \sigma(x)$ and $\mathrm{lca}_T(a',b') \preceq_T w \prec_T v$. Since a' was chosen arbitrarily, we conclude that there cannot be an arc $(a,b) \in E$ such that $\mathrm{lca}_T(a,b) = v$; a contradiction. It follows that there is a support leaf y of vertex v with $\sigma(y) \neq \sigma(x)$. Hence, $\mathrm{lca}_T(x,y) = v \preceq_T \mathrm{lca}_T(x'',y)$ for all $x'' \in L(T)$ with $\sigma(x'') = \sigma(x)$, and thus $(y,x) \in E$ and $y \in N^-(x)$. Since $S_\rho \neq \emptyset$ and $\sigma(y) \notin \sigma(L(T) \setminus L(T(v)))$, there must be a leaf $x' \in S_\rho$ with $\sigma(x') = \sigma(x)$. The fact that $\mathrm{lca}_T(x,y) = v \prec_T \rho = \mathrm{lca}_T(x',y)$ implies $(y,x') \notin E$. Therefore and since $\sigma(x') \neq \sigma(y)$, it follows $y \notin U$. Together with $y \in N^-(x)$, we conclude that $x \notin S^{(1)} = S$.

In summary, we have shown $S = S_\rho$ for any BMG $(\vec{G},\sigma)$. Finally, $S = S_\rho$ together with Cor. 4.15 implies that $S \neq \emptyset$ if and only if $(\vec{G},\sigma)$ is connected, which completes the proof. □

4.4.4 *Construction of LRTs of 2-BMGs*

Thm. 4.8 provides not only a convenient necessary condition for connected 2-BMGs but also a fast way of determining the support set $S = S_\rho$ and thus also a fast recursive approach to construct the LRT for a 2-BMG. It is formalized in Alg. 3 and illustrated in Fig. 21.

Lemma 4.32. *Let $(\vec{G},\sigma)$ be a connected 2-BMG. Then Alg. 3 returns the least resolved tree for $(\vec{G},\sigma)$.*

Proof. Let (T,σ) be the (unique) least resolved tree of $(\vec{G},\sigma)$ with root ρ. The latter is supplied to Alg. 3 to initialize the tree. By Thm. 4.8, Lemma 4.31 and since $(\vec{G},\sigma)$ is connected, the set of support leaves $S_\rho = S^{(2)} = S^{(1)} \neq \emptyset$ for the root ρ is correctly identified in the top-level recursion of Alg. 3 (Line 2-4) and attached to the root ρ (Line 8-9). According to Cor. 4.16, one can now proceed to recursively construct the LRTs for the connected components of $(\vec{G} - S_\rho, \sigma_{|.})$, which is done in Lines 10-15. By Lemma 4.29, these connected components $(\vec{G}_v, \sigma_{|.})$ are exactly the BMGs $\vec{G}(T(v), \sigma_{|.})$ with $v \in \mathsf{child}_T(\rho) \setminus \{S_\rho\}$ (Line 14). In particular, therefore, we have $V(\vec{G}_v) = L(T(v))$. Since $v \notin S_\rho$, i.e., v is an inner vertex, Cor. 4.13 and $v \prec_T \rho$ imply $|\sigma(L(T(v)))| > 1$.

Algorithm 3: LRT for connected 2-colored BMGs $(\vec{G},\sigma)$.

```
   Input: Connected properly 2-colored digraph $(\vec{G}=(L,E),\sigma)$, vertex $\rho$.
   Output: LRT of $(\vec{G},\sigma)$ if $(\vec{G},\sigma)$ is a BMG.
1  Function Build2ColLRT($\vec{G},\sigma,\rho$)
2      $U \leftarrow \{x \in L \mid \mathrm{outdeg}(x) = |L| - |L[\sigma(x)]|\}$        // umbrella vertices
3      $S^{(1)} \leftarrow \{x \in U \mid N^-(x) \subseteq U\}$                // all in-neighbors in $U$
4      $S^{(2)} \leftarrow \{x \in S^{(1)} \mid N^-(x) \subseteq S^{(1)}\}$          // all in-neighbors in $S^{(1)}$
5      if $S^{(1)} = \emptyset$ or $S^{(2)} \neq S^{(1)}$ then
6          exit false
7      else
8          foreach $x \in S^{(2)}$ do
9              add $x$ as a child of $\rho$
10         foreach connected component $\vec{G}_v$ of $\vec{G} - S^{(2)}$ do
11             if $|V(\vec{G}_v)| = 1$ then
12                 exit false
13             create vertex $v$
14             $T_v \leftarrow$Build2ColLRT($\vec{G}_v,\sigma_{|.},v$)
15             connect the root $v$ of $T_v$ as a child to $\rho$
```

Hence, in particular, the condition $|V(\vec{G}_v)| > 1$ (cf. Line 11) to proceed recursively is satisfied for each connected component. ☐

Theorem 4.9. *Given a connected properly 2-colored digraph $(\vec{G},\sigma)$ as input, Alg. 3 returns a tree T if and only if $(\vec{G},\sigma)$ is a 2-colored BMG. In particular, T is the unique least resolved tree for $(\vec{G},\sigma)$.*

Proof. By Lemma 4.32, Alg. 3 returns the unique least resolved tree T if $(\vec{G},\sigma)$ is a connected 2-colored BMG. To prove the converse, suppose that Alg. 3 returns a tree T given the connected properly 2-colored digraph $(\vec{G},\sigma)$ as input. We will show that $(\vec{G},\sigma) = \vec{G}(T,\sigma)$, and thus $(\vec{G},\sigma)$ is a BMG.

It is easy to see that $L(T) = V(\vec{G})$ must hold since, in each step of Alg. 3 every vertex is either attached to some inner vertex or passed down to a deeper-level recursion as part of some connected component. Therefore, every vertex of $\vec{G}$ eventually appears in the output. Thus $\sigma(L(T)) = \sigma(V(\vec{G}))$ and $|\sigma(L(T))| = |\sigma(V(\vec{G}))| = 2$. It remains to show $E(\vec{G}) = E(\vec{G}(T,\sigma))$.

Note first that neither $(\vec{G},\sigma)$ nor $\vec{G}(T,\sigma)$ contain arcs between vertices of the same color. Moreover, since Alg. 3 eventually returns a tree, we have $S^{(1)} = S^{(2)} \neq \emptyset$ in every recursion step. Throughout the remainder of the proof, we will write $S_i^{(1)}$ and $S_i^{(2)}$ for the sets $S^{(1)}$ and $S^{(2)}$ of the i^{th} recursion step. Likewise, in every step, each connected component $(\vec{G}_v,\sigma_{|.})$ computed in Line 10 must contain at least two vertices (cf. Line 11), and thus $|\sigma(V(\vec{G}_v))| = 2$ because $(\vec{G},\sigma)$ is properly 2-colored.

First, let S be the support set of $\vec{G}(T,\sigma)$ and $x \in S$ be arbitrary. Note that the support set is computed in the first iteration step of the algorithm as $S = S_1^{(2)}$, hence $S = S_1^{(2)} \neq \emptyset$. By construction of T, x is attached as a leaf to ρ, i.e. $\mathrm{lca}_T(x,y) = \rho$. Consequently, (x,y) is an arc in $\vec{G}(T,\sigma)$ for all $y \in V(\vec{G})$ with $\sigma(y) \neq \sigma(x)$. By construction of S in Alg. 3, we have $x \in S \subseteq U$, i.e. x is an umbrella vertex in $(\vec{G},\sigma)$ and has out-arcs to every vertex $y \in V(\vec{G})$ with $\sigma(y) \neq \sigma(x)$. Hence, all arcs of the form (x,y) with $x \in S$ and $\sigma(x) \neq \sigma(y)$ exist both in $(\vec{G},\sigma)$ and in $\vec{G}(T,\sigma)$. The latter

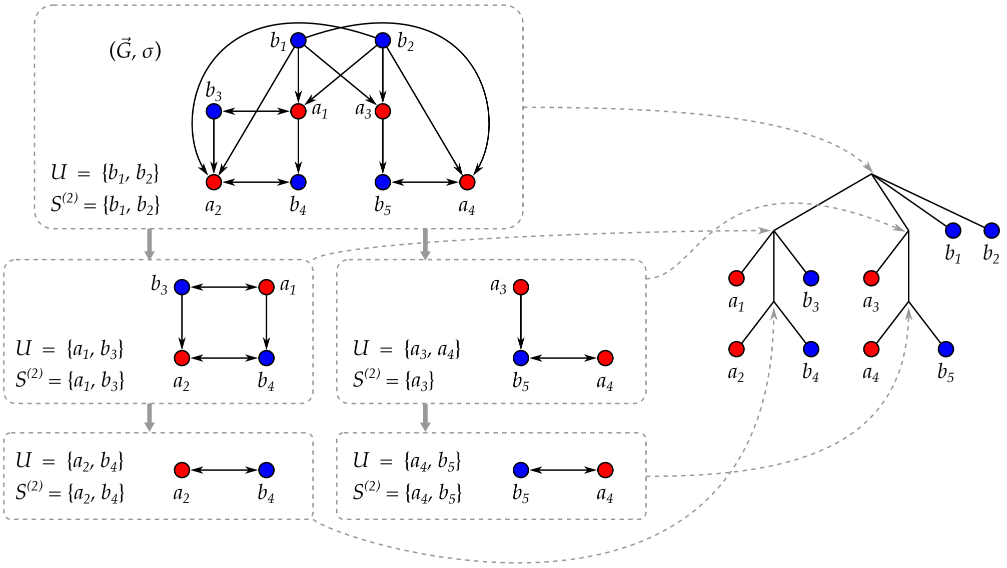


Fig. 21. Illustration of Alg. 3 with input $(\vec{G}, \sigma)$ (uppermost box). The boxes indicate the five recursion steps that are necessary to decompose $(\vec{G}, \sigma)$, and correspond to the five inner vertices of the LRT shown on the right. Note that, in the recursion step on $(\vec{G}[\{a_3, a_4, b_5\}], \sigma_{|.})$, we have $U \neq S^{(2)}$.

property is in particular satisfied for all vertices in S and hence, all arcs between differently colored elements in S exist both in $(\vec{G}, \sigma)$ and in $\vec{G}(T, \sigma)$. Now consider an arbitrary vertex $y \in V(\vec{G}) \setminus S$. Clearly, all in-neighbors in $(\vec{G}, \sigma)$ of the elements in $S = S_1^{(2)}$ must be contained in S, as a consequence of the condition $S_1^{(1)} = S_1^{(2)}$ (cf. Line 5) and the construction of $S_1^{(1)}$ and $S_1^{(2)}$. Hence, $y \notin S$ and $x \in S$ implies that (y, x) is not an arc in $(\vec{G}, \sigma)$. Moreover, $y \notin S$ also implies that y is part of some connected component $(\vec{G}_v, \sigma_{|.})$ of $(\vec{G} - S, \sigma_{|.})$. Therefore, and because Alg. 3 returns T, we must have $y \in V(\vec{G}_v) = L(T(v))$ for some inner vertex $v \in \mathsf{child}_T(\rho)$. As argued above, $(\vec{G}_v, \sigma_{|.})$ and thus also the subtree $T(v)$ contain both colors. Together with Obs. 4.3 and $x \notin L(T(v))$, this implies that $\vec{G}(T, \sigma)$ does not contain the arc (y, x). By the same arguments, there is no arc (y, x') in $\vec{G}(T, \sigma)$ such that the vertex x' is contained in a different connected component $(\vec{G}_{v'}, \sigma_{|.}) \neq (\vec{G}_v, \sigma_{|.})$ of $(\vec{G} - S, \sigma_{|.})$ than y. Since $x \in S$ and $y \notin S$ were chosen arbitrarily, we conclude that (i) any arc incident to some vertex in S exists in $(\vec{G}, \sigma)$ if and only if it exists in $\vec{G}(T, \sigma)$, and (ii) $\vec{G}(T, \sigma)$ contains no arcs between distinct connected components of $(\vec{G} - S, \sigma_{|.})$. Hence, it remains to consider the arcs within a connected component $(\vec{G}_v, \sigma_{|.})$ of $(\vec{G} - S, \sigma_{|.})$.

Alg. 3 recurses on each such connected component $(\vec{G}_v, \sigma_{|.})$ using a newly created vertex $v \in \mathsf{child}_T(\rho)$ to initialize the tree $T(v)$. By Lemma 4.9, it clearly holds that, for any $x, y \in L(T(v)) = V(\vec{G}_v)$, (x, y) is an arc in $\vec{G}(T, \sigma)$ if and only it is an arc in $\vec{G}(T(v), \sigma)$. Thus, it suffices to consider only the subtree $T(v)$. Now, we can apply the same arguments as in the previous recursion step to conclude that all arcs incident to the support set $S_2^{(2)}$ constructed in the current recursion step are the same in $(\vec{G}, \sigma)$ and $\vec{G}(T, \sigma)$ and that neither $(\vec{G}, \sigma)$ nor $\vec{G}(T, \sigma)$ contain arcs between distinct connected components of $(\vec{G}_v - S_2^{(2)}, \sigma_{|.})$. Hence, it suffices to consider the connected components of $(\vec{G}_v - S_2^{(2)}, \sigma_{|.})$. Repeated application of this argumentation results in a chain of connected components that are contained in each other. Since Alg. 3

finally returns a tree, this chain is finite, say with a last element $(\vec{G}_w - S_k^{(2)}, \sigma_{|.})$, and thus $S_k^{(2)} = V(\vec{G}_w)$. In particular, therefore, every vertex in $V(\vec{G})$ is contained in the support set of some recursion step.

In summary, we have shown that $\vec{G}(T,\sigma) = (\vec{G},\sigma)$. Hence, $(\vec{G},\sigma)$ is a connected 2-BMG and, by Lemma 4.32, T is the unique least resolved tree of $(\vec{G},\sigma)$. $\square$

The construction in Lines 2-4 in Alg. 3 naturally produces two cases, $U = S^{(1)} = S^{(2)}$ and $S^{(2)} \subseteq S^{(1)} \subsetneq U$. The following result shows that the latter case implies that the corresponding interior node in the LRT has only a single non-leaf descendant:

Lemma 4.33. *Let $(\vec{G},\sigma)$ be a 2-BMG and S_ρ the support leaves of the root ρ of its LRT (T,σ). If $W := U(\vec{G},\sigma) \setminus S_\rho \neq \emptyset$, then the following statements are true:*

1. *$S_\rho \neq \emptyset$, $\vec{G}$ is connected, and $\vec{G} - S_\rho$ is connected.*
2. *All vertices in $U(\vec{G},\sigma) = S_\rho \,\dot\cup\, W$ have the same color.*
3. *The set of support leaves S_v of the unique inner vertex child v of ρ contains vertices of both colors.*
4. *$W \subsetneq S_v$.*

Proof. First recall that, by Thm. 4.8 and the definition of the support set S of $(\vec{G},\sigma)$, we have $S_\rho = S \subseteq U(\vec{G},\sigma)$, and thus $U(\vec{G},\sigma) = S_\rho \,\dot\cup\, W$. Moreover, by Lemma 4.29, the connected components of $(\vec{G} - S_\rho, \sigma_{|.})$ are exactly the BMGs $\vec{G}(T(v),\sigma_{|.})$ with $v \in \mathsf{child}_T(\rho) \setminus S_\rho$. The vertices $v \in \mathsf{child}_T(\rho) \setminus S_\rho$ are all inner vertices of T since, by definition, the support leaves S_ρ are exactly the children of ρ that are leaves. Together with the contraposition of Lemma 4.26 this implies that $T(v)$ contains both colors.

Statement 1: Let $x \in W$, which exists due to the assumption $W := U(\vec{G},\sigma) \setminus S_\rho \neq \emptyset$. Since $x \notin S_\rho$, it must be part of some connected component of $(\vec{G} - S_\rho, \sigma_{|.})$, say $\vec{G}(T(v),\sigma_{|.})$ for some $v \in \mathsf{child}_T(\rho) \setminus S_\rho$. Now assume, for contradiction, that $\vec{G} - S_\rho$ consists of more than one connected component. By Lemmas 4.29 and 4.26, there is a vertex $v' \in \mathsf{child}_T(\rho) \setminus S_\rho$ such that $v \neq v'$ and both subtrees $T(v)$ and $T(v')$ contain both colors. Hence, there are distinct $y \in L(T(v))$ and $y' \in L(T(v'))$ with $\sigma(y) = \sigma(y') \neq \sigma(x)$. Together with $x \in L(T(v))$, we therefore have $\mathrm{lca}_T(x,y) \preceq_T v \prec_T \rho = \mathrm{lca}_T(x,y')$, which implies $(x,y') \notin E(\vec{G})$. However, $x \in W \subseteq U(\vec{G},\sigma)$ and $\sigma(y') \neq \sigma(x)$ imply $(x,y') \in E(\vec{G})$; a contradiction. Hence, we conclude that $\vec{G} - S_\rho$ has exactly one connected component, and thus ρ has a single inner vertex child v. Since T is phylogenetic, the latter implies that ρ must be incident to at least one leaf, i.e. $S_\rho \neq \emptyset$. Together with Thm. 4.8 this in turn implies that $\vec{G}$ is connected. In summary, Statement 1 is true.

Statement 2: Let $x \in W$ as in the proof of Statement 1. By arguments analogous to those used for Statement 1, we conclude that $\sigma(x) = \sigma(y)$ for every $y \in S_\rho$, since otherwise we would obtain $(x,y) \notin E(\vec{G})$, and thus a contradiction to $x \in U(\vec{G},\sigma)$. Since $x \in W$ was chosen arbitrarily and S_ρ is non-empty, we immediately obtain that all vertices in $U(\vec{G},\sigma) = S_\rho \,\dot\cup\, W$ have the same color, i.e., Statement 2 is true.

Statement 3: Now consider the single inner vertex child v of ρ, and its set of support leaves S_v, which must be non-empty by Lemma 4.28. Note that W must be entirely contained in $L(T(v))$ and recall that all vertices in $S_\rho \,\dot\cup\, W$ are of the same color (cf. Statement 2). First suppose, for contradiction, that S_v only contains vertices of the

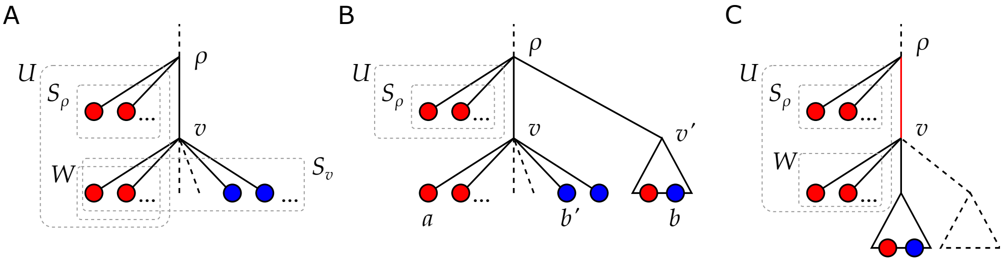


Fig. 22. Illustration of Lemma 4.33. (A) The (local) situation if $W = U \setminus S_\rho \neq \emptyset$ as implied by Lemma 4.33. In particular, ρ only has a single inner vertex child v, all vertices in $U = S_\rho \uplus W$ have the same color, S_v contains vertices of both colors, and $W \subsetneq S_v$. (B) There cannot be a second inner vertex child v', since then none of the vertices except those in S_ρ can be umbrella vertices, e.g. (a,b) is not an arc in the digraph explained by the tree in (B). Hence, this situation is not possible for $W \neq \emptyset$. (C) If S_v does not contain vertices of both colors, then the edge ρv is redundant in the tree, contradicting that (T,σ) in Lemma 4.33 is the LRT.

opposite color as the vertices in $S_\rho \uplus W$. This immediately implies $S_v \cap W = \emptyset$, thus every vertex $x \in W$ must be located in a subtree $T(w)$ of some inner vertex child w of v. Again by contraposition of Lemma 4.26, every such $T(w)$ contains both colors. However, this contradicts $(x,y) \in E(\vec{G})$ for every $y \in S_v$, which must hold as a consequence of $x \in W \subset U(\vec{G},\sigma)$ and $\sigma(y) \neq \sigma(x)$. Next suppose, for contradiction, that S_v only contains vertices of the *same* color as the vertices in $S_\rho \uplus W$. In this case, we obtain that the edge ρv is redundant w.r.t. $(\vec{G},\sigma)$. To see this, consider an arc $(x,y) \in E(\vec{G})$ such that $\mathrm{lca}_T(x,y) = v$. Clearly, x must be directly incident to v, since otherwise the subtree below v to which x belongs would contain both colors, and thus contradict $(x,y) \in E(\vec{G})$. In other words, every such vertex x is a support leaf of v, thus $\sigma(x) = \sigma(S_v) = \sigma(S_\rho)$ and $\sigma(y) \neq \sigma(S_\rho)$. In particular, there exists no arc $(x,y) \in E(\vec{G})$ such that $\mathrm{lca}_T(x,y) = v$ and $\sigma(y) \in \sigma(L(T) \setminus L(T(v))) = \sigma(S_\rho)$ and therefore, by Lemma 4.6, the inner edge ρv is redundant. However, this contradicts the fact that T is least resolved. In summary, only the case in which $S_v \neq \emptyset$ contains vertices of both colors is possible, and thus Statement 3 is true.

Statement 4: First, recall from the proof of Statement 3 that $W \subseteq L(T(v))$ for the single inner vertex child v of ρ. In order to see that $W \subseteq S_v$, assume for contradiction that this is not the case. By similar arguments as used for showing Statement 3, this implies that some $x \in W$ lies in a 2-colored subtree $T(w)$ for some $w \in \mathrm{child}_T(v) \setminus S_v$. Together with the above established fact that S_v contains both colors, this contradicts $x \in U(\vec{G},\sigma)$. Finally, $W \neq S_v$ is a consequence of the fact that S_v contains both colors (Statement 3) but $W \subseteq S_\rho \uplus W$ contains only one color (Statement 2). ☐

We now use this result to consider the performance of Alg. 3.

Lemma 4.34. *Alg. 3 can be implemented to run in $O(|E| \log^2 |V|)$ time for a connected input digraph.*

Proof. Since $\vec{G}$ is connected by assumption, we have $|V| \in O(|E|)$. Starting from $(\vec{G},\sigma)$, the list of out-degrees can be constructed in $O(|E|)$. The initial umbrella set U is then obtained by listing the vertices with maximal out-degree in the color class. The initial set $S^{(1)}$ is constructed by checking, for each $u \in U$, the in-neighbors of u for membership in U in $O(|V| + |E|)$ operations. Then $S^{(2)}$ is obtained in the same manner from $S^{(1)}$, requiring $O(|V| + |E|)$ operations. The initial umbrella set U and the sets $S^{(1)}$ and $S^{(2)}$ thus can be constructed in linear time. In each recursive call of `Build2ColLRT`, at least one leaf is split off, hence the recursion depth is $|V| - 1$ in the worst case. Since the support vertices removed in each step have all of their in-neighbors in U, their removal does not affect the out-neighborhood for any $x \in V(\vec{G} - U) \subseteq V(\vec{G} - S^{(2)})$, and hence, $\mathsf{outdeg}(x)$ does not require updates. The in-neighborhoods $N^-(x)$ can be updated by removing the arcs between $\vec{G} - S^{(2)}$ and $S^{(2)}$ as a consequence of Lemma 4.29 and Thm. 4.8. Since every arc appears exactly once in the removal, the total effort for these updates is $O(|E|)$.

We continue by showing that every vertex needs to be considered as an umbrella vertex at most twice, and that the total effort of constructing all sets $S^{(1)}$ and $S^{(2)}$ is $O(|E|)$, given that the umbrella vertices U can be obtained efficiently, which we discuss afterwards. To this end, we distinguish, for each of the single recursion steps, two cases: $S^{(1)} = U$ and $S^{(1)} \subsetneq U$. First if $S^{(1)} = U$, and thus also $S^{(2)} = S^{(1)} = U$, we consider each in-arc of $x \in U$. Since these vertices and their corresponding arcs are removed when constructing $\vec{G} - S^{(2)}$, they are not considered again in a deeper recursion step. In the second case, we have $S^{(1)} \subsetneq U$, which together with $S^{(2)} = S^{(1)}$ implies $W := U \setminus S^{(2)} \neq \emptyset$, and only the vertices in $U \setminus W$ are removed. However, Lemma 4.33 guarantees that, for a 2-BMG as input digraph, the vertices in W appear as support leaves in the next step and thus appear in the update of U, $S^{(1)}$, and $S^{(2)}$ no more than a second time. In order to use the properties in Lemma 4.33 for the general case (i.e. $(\vec{G}, \sigma)$ is not necessarily a BMG), we can, whenever $W \neq \emptyset$, (i) check that $\vec{G} - S^{(2)}$ only has a single connected component $\vec{G}_v$, and (ii) pass down the set W to the recursion step on $\vec{G}_v$ in which the condition $W \subsetneq S^{(2)}$ is checked. If any of these checks fails, we can exit false. This way, we ensure that every vertex appears at most two times as an umbrella vertex in the general case. To construct $S^{(1)}$ from U, we have to scan the in-neighborhood $N^-(x)$ of each vertex $x \in U$ and check whether $N^-(x) \subset U$. We repeat this step to construct $S^{(2)}$ from $S^{(1)}$. Membership in U and $S^{(1)}$, resp., can be checked in constant time (e.g. by marking the vertices in the current set U). Since we have to consider each vertex, and hence, each in-neighborhood at most twice, all sets $S^{(1)}$ and $S^{(2)}$ can be obtained with a total effort of $O(|E|)$.

It remains to show that the input digraph can be decomposed efficiently in such a way that the connectivity information is maintained and the candidates for umbrella vertices in each component are updated. The connected components $\vec{G}_v$ can be obtained by using the dynamic data structure described in [146], often called HDT data structure. It maintains a maximal spanning forest representing the underlying undirected graph with edge set $\widetilde{E} = \{xy \mid (x,y) \in E \text{ or } (y,x) \in E\}$, and allows deletion of all $|\widetilde{E}| \in O(|E|)$ edges with amortized cost $O(\log^2 |V|)$ per edge deletion. The explicit traversal of the connected components to compute U can be avoided as follows: Since $\mathsf{outdeg}(x)$ does not require updates, we can maintain a doubly-linked list of vertices x for each color $i \in \{1,2\}$, and each value of $\mathsf{outdeg}(x)$ where $\sigma(x) = i$. In order to be able to access the highest value of the out-degrees, we maintain these values together with pointer to the respective doubly-linked list in balanced binary search trees (BST), one for each color and each connected component. The BSTs for the two colors are computed first for $(\vec{G}, \sigma)$ in $O(|V| \log(|V|))$ time and afterwards updated to fit with the out-degree of the currently considered component $\vec{G}_v$. To update these lists and BSTs for $\vec{G}_v$, observe first that $\vec{G}_v$ can be obtained from $\vec{G}$ by stepwise deletion of single arcs, i.e. edges in the HDT data structure representing the underlying undirected versions. We update, resp., construct the pair of BSTs (one for each color) for each connected component as follows: Since a single arc deletion splits a connected component $\vec{G}'$ into at most two connected components $\vec{G}_1$, and $\vec{G}_2$, we can apply the well-known technique of traversing the smaller component [280]. The size of each connected component can be queried in $O(1)$ time in the HDT data structure. Suppose w.l.o.g. that $|V(\vec{G}_1)| \leq |V(\vec{G}_2)|$. We construct a new pair of BSTs for $\vec{G}_1$, and delete the vertices $V(\vec{G}_1)$ and the respective degrees from the two original

BSTs for $\vec{G}$, which then become the BSTs for $\vec{G}_2$. More precisely, we delete each vertex $x \in V(\vec{G}_1)$ in the respective list corresponding to $\texttt{outdeg}(x)$, and if the length of this list drops to zero, we also remove the corresponding out-degree in the BST. Likewise, we insert the out-degree of x and an empty doubly-linked list into the newly-created BST for $\vec{G}_1$, if it is not yet present, and append x to this list. Note that the number of out-degree deletions and insertions does not exceed $|V(\vec{G}_1)|$. Due to the technique of traversing the smaller component, every vertex is deleted and inserted at most $\lfloor \log |V| \rfloor$ times. Therefore, we obtain an overall complexity of $O(|V| \log^2 |V|)$ for the maintenance of the BSTs where the additional log-factor originates from re-balancing the BSTs whenever necessary.

In each recursion step, the set U can now be obtained by listing (at most) the vertices with the maximal out-degree for each of the two colors. Finding the two out-degrees and corresponding lists in the BSTs requires $O(\log |V|)$ in each step, and thus $O(|V| \log |V|)$ in total. In order to determine whether these candidates x are actually umbrella vertices, we have to check whether $\texttt{outdeg}(x) = |V(\vec{G}_v)| - |V(\vec{G}_v)[\sigma(x)]|$. The HDT data structure allows constant-time query of the size of a given connected component, since this information gets updated during the maintenance of the spanning forest. By the same means, we can keep track of the number of vertices of a specific color in each connected components. Note that we only need to do this for one color r since $|V(\vec{G}_v)[s]| = |V(\vec{G}_v)| - |V(\vec{G}_v)[r]|$. This does not increase the overall effort for maintaining the data structure since it happens alongside the update of $|V(\vec{G}_v)|$.

In summary, the total effort is dominated by maintaining the connectedness information while deleting $O(|E|)$ arcs, i.e., $O(|E| \log^2 |V|)$ time. ☐

As a direct consequence of Thm. 4.8, the LRT of a disconnected digraph $\vec{G}$ is obtained by connecting the roots of the LRTs T_v of the connected components $\vec{G}_v$ to an additional root vertex, see also [102, Cor. 4]. Lemma 4.34 thus implies

Theorem 4.10. *The LRT of a 2-BMG can be computed in* $O(|V| + |E| \log^2 |V|)$.

Proof. The connected components $\vec{G}_i = (V_i, E_i)$ of $\vec{G} = (V, E)$ can be enumerated in $O(|V| + |E|)$ operations, e.g. using a breadth-first search on the underlying undirected graph. By Lemma 4.34, $O(|E_i| \log^2 |V_i|) \leq O(|E_i| \log^2 |V|)$ operations are required for each $\vec{G}_i$. Hence, the total effort is $O(|V| + |E| + \log^2 |V| \sum_i |E_i|) = O(|V| + |E| \log^2 |V|)$. ☐

In order to illustrate the improved complexity for the construction of LRTs of 2-BMGs, we implemented both the triple-based approach in Alg. 1, i.e., the application of `BUILD` [7] with the informative triples defined in Eq. (3) as input, and Alg. 3. As input, we used 2-BMGs that were randomly generated as follows: First, we simulate random trees T recursively, starting from a single vertex, by attaching to a randomly chosen vertex v either a single leaf if v is an inner vertex of T or a pair of leaves if v is a leaf. The construction stops when the desired number of leaves is reached. Note that the resulting tree is phylogenetic by construction. Each leaf is then colored by selecting at random one of the two colors. Finally, we compute the 2-BMG $\vec{G}(T, \sigma)$ from each of the simulated leaf-colored trees (T, σ).

Both methods for the LRT computation were implemented in Python. Moreover, we note that we did not implement the sophisticated dynamic data structures used in the proof of Lemma 4.34, but a rather naïve implementation of Alg. 3. Similarly, we used a simple implementation of `BUILD` instead

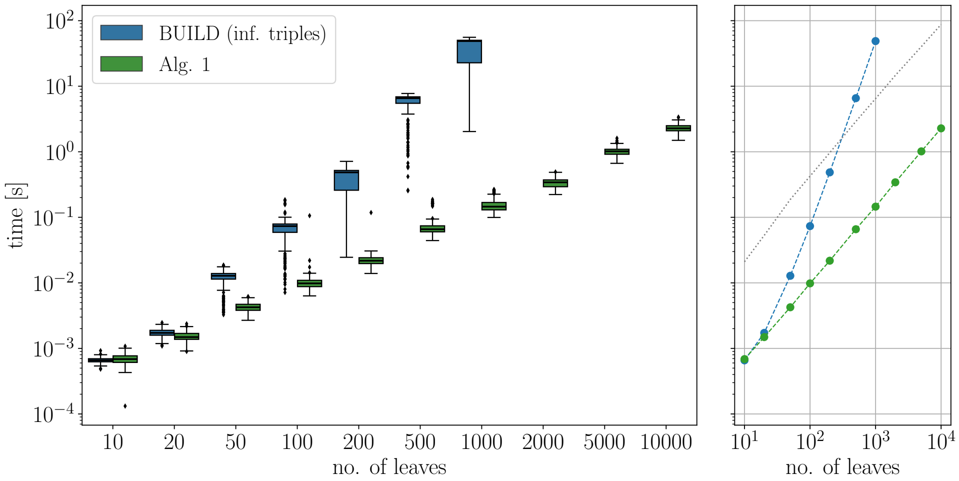


Fig. 23. Running time comparison of the general approach for constructing an LRT using Alg. 1, i.e. `BUILD`, (blue) vs. Alg. 3 (green). For each number of leaves, 200 2-BMGs were generated as described in the text. In the right panel, the median values are shown with logarithmic axes. The additional dotted line indicates the median values of the size of the simulated BMGs, i.e. the number of arcs, scaled by a factor 10^{-3}. We did not compute the LRTs with the first method for instances with more than 1000 leaves because of the excessive computational cost.

of the $O(|V|\,|E| \log^2 |V|)$ version of the approach for ℓ-BMGs detailed in [102] and Sec. 4.2.3. In this setting, Fig. 23 shows a remarkable improvement of the running time when compared to the general approach. Empirically, we observe that the running time of Alg. 3 indeed scales nearly linearly with the number of edges.

The decomposition of 2-BMGs based on support vertices may provide an avenue for a class of heuristic algorithms that correct a given 2-colored digraph, that was estimated from real-life data and thus contains noise, to a 2-BMG by recursively identifying candidates for support leaves in the underlying "true" LRT. This is conceptually distinct from the approaches for BMG editing that are based on triples $\mathcal{R}(\vec{G},\sigma)$ or $\mathcal{R}^{\mathrm{B}}(\vec{G},\sigma)$ as presented in Sec. 5.5 later in this work; however, restricted to 2-colored digraphs.

4.4.5 *Binary-Explainable 2-BMGs*

In this section, we briefly describe a modification of Alg. 3 that allows the efficient recognition of binary-explainable 2-BMGs.

The following Lemma shows that the third statement in Prop. 4.9 can be translated to a much simpler statement in terms of the support leaves of its LRT.

Lemma 4.35. *A 2-BMG $(\vec{G},\sigma)$ contains an induced hourglass if and only if its LRT (T,σ) contains an inner vertex u such that S_u contains support vertices of both colors and $V(\vec{G}(T(u)) - S_u) \neq \emptyset$.*

Proof. By Thm. 4.9, Alg. 3 returns the LRT (T, σ) for $(\vec{G}, \sigma)$ if and only if $(\vec{G}, \sigma)$ is a 2-BMG. Hence, we assume in the following that the latter is satisfied. As a consequence of Prop. 4.9 and the fact that (T, σ) explains $(\vec{G}, \sigma)$, we know that $(\vec{G}, \sigma)$ is binary-explainable if and only if there is no vertex $u \in V^0(T)$ with three distinct children v_1, v_2, and v_3 and two distinct colors r and s satisfying (a) $r \in \sigma(L(T(v_1)))$, $r, s \in \sigma(L(T(v_2)))$, and $s \in \sigma(L(T(v_3)))$, and (b) $s \notin \sigma(L(T(v_1)))$, and $r \notin \sigma(L(T(v_3)))$.

First, suppose that $(\vec{G}, \sigma)$ contains an hourglass, i.e., by Prop. 4.9 there is a vertex $u \in V^0(T)$ with distinct children v_1, v_2, and v_3 and two distinct colors r and s satisfying (a) and (b). Since $(\vec{G}, \sigma)$ is 2-colored and (T, σ) is its LRT, Lemma 4.26 together with $s \notin \sigma(L(T(v_1)))$ and $r \notin \sigma(L(T(v_3)))$ implies that v_1 of color r and v_2 of color s, respectively, are both leaves. In particular, therefore, we know that $v_1, v_2 \in S_u$ are support leaves. By Lemma 4.29 and since $\vec{G}(T(u), \sigma_{|.})$ is also a BMG, the connected components of $(\vec{G}(T(u)) - S_u, \sigma_{|.}) = (\vec{G}[L(T(u))] - S_u, \sigma_{|.})$ (cf. Lemma 4.9) are exactly the BMGs $\vec{G}(T(v), \sigma_{|.})$ with $v \in \mathsf{child}_T(u) \setminus S_u$. Together with the fact that $v_2 \in V^0(T)$ as a consequence of $L(T(v_2))$ containing both colors r and s, this implies that $(\vec{G}(T(u)) - S_u, \sigma_{|.})$ is not the empty digraph.

Conversely, suppose there is a vertex $u \in V^0(T)$ such that S_u contains support vertices v_1 and v_3 with distinct colors $\sigma(v_1) \neq \sigma(v_3)$ and $V(\vec{G}(T(u)) - S_u) \neq \emptyset$, i.e., u has a child $v_2 \in \mathsf{child}_T(u) \setminus S_u$ that is not a support leaf and hence satisfies $v_2 \in V^0(T)$. Lemma 4.26 implies that $L(T(v_2))$ contains both colors since $v_2 \in V^0(T)$. Hence, the three children v_1, v_2, and v_3 of u satisfy conditions (a) and (b) of Prop. 4.9(3), and thus $(\vec{G}, \sigma)$ contains an induced hourglass. □

Corollary 4.17. *It can be checked in $O(|V| + |E| \log^2 |V|)$ whether or not a properly 2-colored digraph $(\vec{G}, \sigma)$ is a binary-explainable BMG.*

Proof. Recall that there is a one-to-one correspondence between the recursion step in Alg. 3 and the inner vertices $u \in V^0(T)$. As argued in the proof of Lemma 4.34, every vertex appears at most twice in an umbrella set U. Therefore, it can be checked in $O(|V|)$ total time whether $S = S^{(2)}$ contains vertices of both colors. Since the vertex set of $\vec{G}_u - S_u$ is maintained in the dynamic graph HDT data structure, it can be checked in constant time for each u whether $\vec{G}_u - S_u$ is non-empty. The additional effort to check the condition of Lemma 4.35 is therefore only $O(|V|)$. Hence, we still require a total effort of $O(|V| + |E| \log^2 |V|)$ (cf. Thm. 4.10). □

Cor. 4.17 improves the complexity for the decision whether a 2-BMG is binary-explainable as compared to the $O(|V|^3 \log^2 |V|)$-time algorithm for (general) BMGs presented in Sec. 4.3.

4.5 SUMMARY

In this chapter, we extensively revisited best match graphs. In particular, a correction of the previously published characterization of BMGs based on informative triples was presented in Thm. 4.2. This will be the starting point for the development of a group of heuristic algorithms for BMG editing in Sec. 5.5. We also gave an alternative characterization using informative and forbidden triples that can be easily extracted from a colored digraph (Thm. 4.3). This will be useful for phrasing and solving the BMG editing problem as an ILP (cf. Sec. 5.4).

We introduced binary-explainable BMGs (beBMGs) as an interesting subclass of BMGs. These are of practical relevance since polytomies in gene trees are often considered to be the result of insufficient data. They can be characterized as hourglass-free BMGs (Prop 4.8), or, alternatively, as sf-colored digraphs $(\vec{G} = (V, E), \sigma)$ for which the triple set $\mathcal{R}^{\mathrm{B}}(\vec{G}, \sigma)$ is consistent (Thm. 4.4). As a consequence of the latter, beBMGs can be recognized in $O(|V|^3 \log^2 |V|)$ (Cor. 4.10). In particular, beBMGs are explained by the unique binary-resolvable tree (BRT) which is often much better resolved than the LRT, and thus, may be a valuable source of information in the application to gene tree reconstruction. We will continue to keep an eye on beBMGs when we turn to BMG modification problems in the next chapter.

We also gave a novel characterization of 2-BMGs in terms of forbidden induced subgraphs, which will turn out to be useful for deriving complexity results and ILP formulations for BMG modification problems in the next chapter. Moreover, we have shown that 2-BMGs have a recursive structure that is reflected in certain induced subgraphs that correspond to subtrees of the LRT. The leaves connected directly to the root of a given subtree play a special role as support vertices in the corresponding subgraph of the 2-BMG. Since the support vertices of the root can be identified efficiently in a given input digraph, there is a recursive decomposition of $(\vec{G}, \sigma)$ that directly yields the LRT, and can be implemented to run in $O(|V| + |E| \log^2 |V|)$ (Alg. 3 and Thm. 4.10) with help of a dynamic data structure to maintain connectedness information [146]. This provides a considerable speed-up compared to the previously known $O(|V||E| \log^2 |V|)$ and $O(|V|^3)$ algorithms [102]. Empirically, we observe a substantial speed-up even if simpler data structures are used to implement Alg. 3.

5

BEST MATCH GRAPH EDITING

Empirically, best matches are estimated on the basis of gene sequence similarities or distances [223], which in turn are usually obtained from sequence alignments [50]. In practice, fast heuristic algorithms such as `BLAST` [17] and its more modern successors are usually used for this purpose [141, 216]. Apart from simple measurement inaccuracies, systematic biases such as deviations from a common molecular clock, i.e., differences in the evolutionary rates of different genes, cause discrepancies between *best hits* (most similar sequences) and *best matches* (evolutionary most closely related sequences). While some systematic effects can be corrected directly (cf. Sec. 4.1.4 and [293]), a residual level of error is unavoidable. It is therefore a question of considerable practical interest in computational biology whether the mathematical properties characterizing BMGs can be used to correct empirical estimates. Formally, this question amounts to a graph editing problem: Given a vertex-colored directed graph $(\vec{G},\sigma)$, what is the minimal number of arcs that need to be inserted or deleted to convert $(\vec{G},\sigma)$ into a BMG $(\vec{G}^*,\sigma)$?

This chapter is organized as follows: In Sec. 5.1, we formally introduce various modification problems for BMGs. We prove in Sec. 5.2 that 2-BMG Deletion and 2-BMG Editing are NP-complete by reduction from Exact 3-Cover, and that 2-BMG Completion is NP-complete by reduction from Chain Graph Completion. These results are used in Sec. 5.3 to establish NP-completeness for any fixed number $\ell \geq 2$ of colors. We provide ILP solutions for the respective ℓ-BMG modification problems in Sec. 5.4. Most of the complexity and ILP results appeared previously in [266]. However, a slightly modified reduction [260] from Exact 3-Cover is presented here which also shows NP-hardness of editing a digraph to a beBMG. The polynomial-time solution for completing a 2-BMG to a beBMG has been published in [261]. We turn to an investigation of several heuristic algorithms for BMG editing in Sec. 5.5. This last part of the chapter is based on [263].

5.1 BMG MODIFICATION PROBLEMS

Best matches are closely linked to the identification of orthologous genes, and considerable efforts have already been expended to devise methods for orthology assessment, see Sec. 3.3.3. The orthology graph of a gene family (with the genes as vertices and undirected edges between orthologous genes) can be shown to be a subgraph of the reciprocal best match graph (RBMG), i.e., the symmetric part of the BMG [103]. This has sparked interest in a characterization of RBMGs [104] and the corresponding graph editing problems [137]. The deletion and the editing problems of 2-colored RBMGs are equivalent to Bicluster Deletion and Bicluster Editing, respectively, a fact that was used to demonstrate NP-hardness for the general,

ℓ-colored case. On the other hand, orthology graphs are cographs [132]. Cograph Editing or Cograph Deletion thus have been used to correct empirical approximations of RBMGs to orthology graphs in [133]. Several related problems have been discussed in the literature, often aiming at using additional biological information as part of the graph modification process, cf. eg. [79, 136, 180, 182, 184, 220]. Both Cograph Editing and Cograph Deletion are NP-complete [197].

In Chapter 6, we will see that knowledge of the BMG makes it possible to identify, in polynomial time, a certain set of edges of the RBMG that cannot be part of the orthology graph and that these edges, in general, do not form an optimal solution of either Cograph Editing or Cograph Deletion. This observation suggests correcting the empirical similarity data at the outset by editing them to the nearest BMGs instead of operating on an empirical approximation of the RBMG.

In real-life applications, we have to expect that digraphs estimated from empirical best match data will contain both erroneous and/or missing arcs. We will assume, however, that the coloring σ of a given digraph $(\vec{G},\sigma)$, encoding the information in which species/genome a gene resides, is correct. Formally, we consider the following graph modification problems for properly colored digraphs.

Problem 5.1 (ℓ-BMG Deletion)**.**

Input:	*A properly ℓ-colored digraph $(\vec{G}=(V,E),\sigma)$ and an integer k.*
Question:	*Is there a subset $F\subseteq E$ such that $\vert F\vert\leq k$ and $(\vec{G}-F,\sigma)$ is an ℓ-BMG?*

It is worth noting that ℓ-BMG Deletion does not always have a feasible solution. In particular, if $(\vec{G},\sigma)$ contains a sink, no solution exits for any $\ell>1$ as a consequence of Thm. 4.3 and the fact that we only delete arcs. In contrast, it is always possible to obtain a BMG from a properly colored digraph $(\vec{G},\sigma)$ if arc insertions are allowed. To see this, observe that the digraph $(\vec{G}',\sigma)$ with $V(\vec{G}')=V(\vec{G})$ that contains all arcs between vertices of different colors is a BMG, since it is explained by the star tree with leaf set $V(\vec{G}')$. This suggests that the following two problems are more relevant for practical applications:

Problem 5.2 (ℓ-BMG Editing)**.**

Input:	*A properly ℓ-colored digraph $(\vec{G}=(V,E),\sigma)$ and an integer k.*
Question:	*Is there a subset $F\subseteq(V\times V)_{irr}$ such that $\vert F\vert\leq k$ and $(\vec{G}\triangle F,\sigma)$ is an ℓ-BMG?*

Problem 5.3 (ℓ-BMG Completion)**.**

Input:	*A properly ℓ-colored digraph $(\vec{G}=(V,E),\sigma)$ and an integer k.*
Question:	*Is there a subset $F\subseteq(V\times V)_{irr}\setminus E$ such that $\vert F\vert\leq k$ and $(\vec{G}+F,\sigma)$ is an ℓ-BMG?*

In Sec. 4.3, we have characterized the subclass of BMGs that can be explained by binary trees. In particular, recall that, by Prop. 4.8, binary-explainable BMGs are exactly the hourglass-free BMGs. As argued before, they are of practical significance because phylogenetic trees are often assumed to be binary by nature, with multifurcations arising in many cases as an artifact of insufficient data [73, 144, 201].

We therefore consider the slightly different modification problems that, given an arbitrary properly colored digraph $(\vec{G},\sigma)$, aim to find a binary-explainable BMG:

Problem 5.4 (ℓ-BMG Editing restricted to Binary-Explainable Graphs (ℓ-BMG EBEG))**.**

Input:	*A properly ℓ-colored digraph $(\vec{G}=(V,E),\sigma)$ and an integer k.*
Question:	*Is there a subset $F\subseteq(V\times V)_{irr}$ such that $\lvert F\rvert\le k$ and $(\vec{G}\bigtriangleup F,\sigma)$ is a binary-explainable ℓ-BMG?*

The corresponding completion and deletion problems will be called ℓ-BMG CBEG and ℓ-BMG DBEG, respectively.

5.2 COMPLEXITY OF 2-BMG MODIFICATION PROBLEMS

In this section, we consider decision problems related to modifying 2-colored digraphs. The general case with an arbitrarily large number $\ell\ge 2$ of colors will be the subject of the following section. For $\ell=2$, we will show that both 2-BMG Editing and 2-BMG Deletion as well as ℓ-BMG EBEG and ℓ-BMG DBEG are NP-complete by reduction from the Exact 3-Cover problem (X3C), one of Karp's famous 21 NP-complete problems [165].

Problem 5.5 (Exact 3-Cover (X3C))**.**

Input:	*A set $\mathfrak{S}$ with $\lvert\mathfrak{S}\rvert=3t$ elements and a collection $\mathcal{C}$ of 3-element subsets of $\mathfrak{S}$.*
Question:	*Does $\mathcal{C}$ contain an exact cover for $\mathfrak{S}$, i.e., a subcollection $\mathcal{C}'\subseteq\mathcal{C}$ such that every element of $\mathfrak{S}$ occurs in exactly one member of $\mathcal{C}'$?*

An exact 3-cover $\mathcal{C}'$ of $\mathfrak{S}$ with $\lvert\mathfrak{S}\rvert=3t$ is necessarily of size $\lvert\mathcal{C}'\rvert=t$ and satisfies $\bigcup_{C\in\mathcal{C}'}C=\mathfrak{S}$.

Theorem 5.1. [165] X3C *is NP-complete.*

In the following, we will make extensive use of properly 2-colored digraphs that contain all possible arcs:

Definition 5.1. *A* bi-clique *of a colored digraph $(\vec{G},\sigma)$ is a subset of vertices $C\subseteq V(\vec{G})$ such that (i) $\lvert\sigma(C)\rvert=2$ and (ii) $(x,y)\in E(\vec{G}[C])$ if and only if $\sigma(x)\neq\sigma(y)$ for all $x,y\in C$. A colored digraph $(\vec{G},\sigma)$ is a* bi-cluster graph *if all its connected components are bi-cliques.*

In a bi-clique, all arcs between vertices of different color are present. Thus, a bi-clique with n and m vertices in the two color classes has $2nm$ arcs, see Fig. 24 for the case $n=m=3$. We emphasize that, in contrast to the definition used in [137], single vertex graphs are not considered as bi-clique.

We start with a simple construction of a special subclass of BMGs that contain a number of bi-cliques and will appear as connected components in our reduction from X3C:

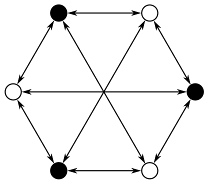


Fig. 24. A (sub)graph induced by a bi-clique consisting of 3 black and 3 white vertices. It has 18 arcs in total.

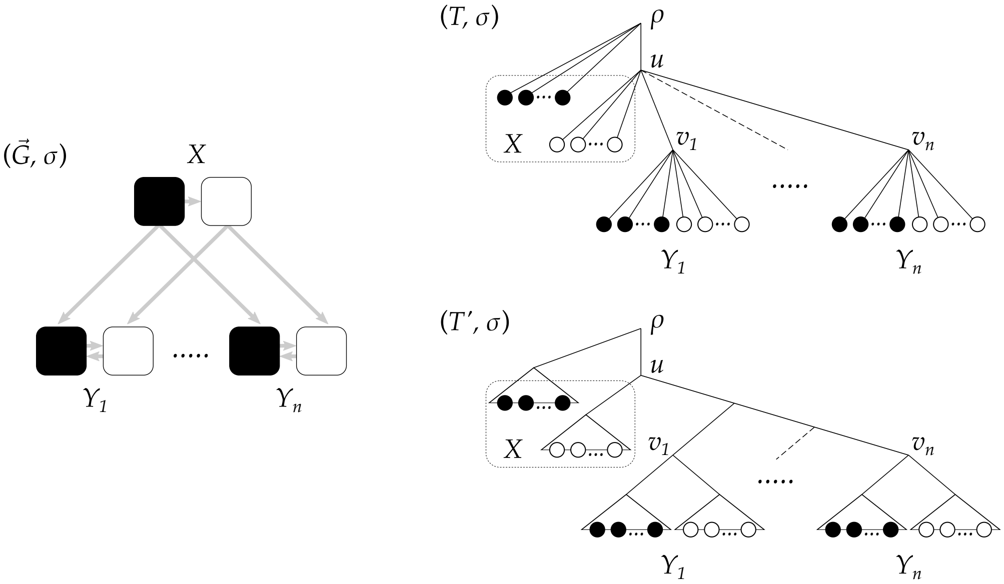


Fig. 25. Illustration of the digraph $(\vec{G}, \sigma)$ constructed in Lemma 5.1 and its (least resolved) tree (T, σ). The boxes represent the set of black and white vertices contained in the sets $X, Y_1, \ldots, Y_n$. The thick gray arrows indicate that all arcs in that direction exist between the respective sets. The tree (T', σ) (with the triangles representing arbitrary binary subtrees) is a possible binary refinement of (T, σ) that explains $(\vec{G}, \sigma)$.

Lemma 5.1. *Let $(\vec{G} = (V, E), \sigma)$ be a 2-colored digraph obtained as follows: Set $V := X \mathbin{\dot\cup} Y_1 \mathbin{\dot\cup} \ldots \mathbin{\dot\cup} Y_n$ where each set in $\mathfrak{C} := \{X, Y_1, \ldots, Y_n\}$, $n \geq 1$, consists of at least one* black *and at least one* white *vertex. For the (initially empty) arc set E, add*

(i) all arcs from *the black vertices in X* to *the white vertices in X,*

(ii) all arcs (x, y) with $x \in X$ and $y \in V \setminus X$ for which $\sigma(x) \neq \sigma(y)$, and

(iii) all arcs (y_1, y_2) such that y_1 and y_2 are contained in the same set $Y_i \in \{Y_1, \ldots, Y_n\}$ and $\sigma(y_1) \neq \sigma(y_2)$.

Then $(\vec{G}, \sigma)$ is a binary-explainable BMG. Moreover, the disjoint union of such digraphs (with the same two colors) is a binary-explainable BMG.

Proof. The construction of the digraph $(\vec{G}, \sigma)$ is illustrated on the l.h.s. of Fig. 25. To show that $(\vec{G}, \sigma)$ is a BMG, it suffices to verify that the tree (T, σ) in Fig. 25 explains $(\vec{G}, \sigma)$. For all black vertices in X, we have $\operatorname{lca}_T(x, y) = \rho$ for all white vertices $y \in V$. Hence, every white vertex $y \in V$ is a best match of every black vertex in X. All white vertices in $x \in X$ are children of u and $\operatorname{lca}_T(x,y) = u$ for all vertices $y \in V \setminus X$. Taken together these two facts imply that every black vertex $y \in V \setminus X$ is a best match of x. Since $n \geq 1$, there is at least one such black vertex $y \in V \setminus X$ and $\operatorname{lca}_T(x,y) = u \prec_T \rho = \operatorname{lca}_T(x,y')$ holds for every black vertex $y' \in X$. Therefore, none of the black vertices in X is a best match of any white vertex in X. If $x,y \in Y_i \in \mathfrak{C} \setminus \{X\}$ and $x', y' \in L(T)$ are all distinct, we have $\operatorname{lca}_T(x,y) = v_i \preceq_T \operatorname{lca}_T(x,y'), \operatorname{lca}_T(x',y)$. Hence every $Y_i \in \mathfrak{C} \setminus \{X\}$ is a bi-clique. Furthermore, if $x' \in L(T) \setminus Y_i$ or $y' \in L(T) \setminus Y_i$, we have $\operatorname{lca}_T(x,y) = v_i \prec_T \operatorname{lca}_T(x',y) \in \{u,\rho\}$ and $\operatorname{lca}_T(x,y) = v_i \prec_T \operatorname{lca}_T(x,y') \in \{u,\rho\}$, resp., and therefore there are no arcs from vertices in Y_i to vertices in X and no arcs between distinct vertex sets $Y_i, Y_j \in \mathfrak{C} \setminus \{X\}$. Therefore, $\vec{G}(T,\sigma) = (\vec{G},\sigma)$, and thus $(\vec{G},\sigma)$ is a BMG.

It is now an easy task to verify that none of the inner vertices of (T,σ) satisfies Condition (a) and (b) in Prop. 4.9(3). Since (T,σ) explains $(\vec{G},\sigma)$, Prop. 4.9 implies that $(\vec{G},\sigma)$ is also binary-explainable.

It remains to show that the disjoint union $(\vec{G}',\sigma')$ of such digraphs $(\vec{G}_i,\sigma_i)$ with the same two colors is a binary-explainable BMG. Since all $(\vec{G}_i,\sigma_i)$ are in particular BMGs, Prop. 1 in [102] implies that $(\vec{G}',\sigma')$ is a BMG. By Prop. 4.8 and since every $(\vec{G}_i,\sigma_i)$ is binary-explainable, every $(\vec{G}_i,\sigma_i)$ is hourglass-free. Since hourglasses are connected, their disjoint union $(\vec{G}',\sigma')$ is also hourglass-free. Applying Prop. 4.8 again, we conclude that $(\vec{G}',\sigma')$ is a binary-explainable BMG. □

We note that the LRT used in the proof of Lemma 5.1 is in general not binary. As argued above, this does not imply that its BMG $(\vec{G},\sigma)$ is not binary-explainable. The tree (T',σ) in Fig. 25 shows a possible binary refinement of the LRT (T,σ).

We are now in the position to prove NP-completeness of 2-BMG EDITING. The strategy of the NP-hardness proof is very similar to the one used in [85] and [197] to show the NP-hardness of COGRAPH EDITING. Nevertheless, although similar in fashion, our proof has subtle but important differences when compared to the proofs provided in [85] and [197]. In particular, at the heart of our construction are 2-colored bi-cliques rather than complete graphs.

Theorem 5.2. 2-BMG EDITING *and* 2-BMG EBEG *are NP-complete.*

Proof. Since BMGs can be recognized in polynomial time by Cor. 4.6, the 2-BMG EDITING problem is clearly contained in NP. Since binary-explainable BMGs can be recognized in polynomial time by Cor. 4.10, the 2-BMG EBEG problem is also contained in NP. To show the NP-hardness, we use reduction from X3C.

Let $\mathfrak{S}$ with $|\mathfrak{S}| = n = 3t$ and $\mathcal{C} = \{C_1, \dots, C_m\}$ be an instance of X3C. Clearly, if $m = t$ the X3C problem becomes trivial and thus, we assume w.l.o.g. that $m > t$. The latter implies that every solution $\mathcal{C}'$ of X3C satisfies $\mathcal{C}' \subsetneq \mathcal{C}$. Moreover, we assume w.l.o.g. that $C_i \neq C_j$, $1 \leq i < j \leq m$. We construct an instance $(\vec{G} = (V,E), \sigma, k)$, where $(\vec{G},\sigma)$ is colored with the two colors black and white, of the 2-BMG EDITING or 2-BMG EBEG problem as follows: First, we construct a bi-clique S consisting of a black vertex s_b and a white vertex s_w for every $s \in \mathfrak{S}$. Thus the subgraph induced by S has $6t$ vertices and $r := 18t^2$ arcs in total. Let $q := 3 \times [6r(m-t) + r - 18t]$. For each of the m subsets C_i in $\mathcal{C}$, we introduce two vertex sets X_i and Y_i, where X_i consists of r black and r white new vertices, and Y_i consists of q black and q white new vertices. Arcs are inserted *from* the black vertices *to* the white vertices in X_i, and all Y_i are completed to bi-cliques. In addition, we insert the following arcs:

- (x,y) for every $x \in X_i$ and $y \in Y_i$ with $\sigma(x) \neq \sigma(y)$ (note $(y,x) \notin E$),

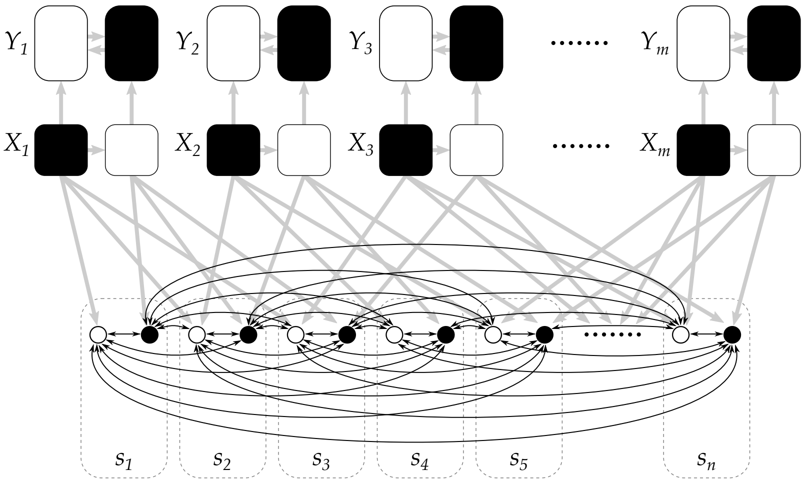


Fig. 26. Illustration of the reduction from Exact 3-Cover. The thick gray arrows indicate that all arcs from that set to another set/vertex exist. The illustration emphasizes the analogy to [85] and [197].

– (x, s_b) for every white vertex $x \in X_i$ and every element $s \in C_i$, and,
– (x, s_w) for every black vertex $x \in X_i$ and every element $s \in C_i$.

This construction is illustrated in Fig. 26. Clearly, $(\vec{G}, \sigma)$ is properly 2-colored and can be constructed in polynomial time.

We set $k := 6r(m-t) + r - 18t$ and show that there is a t-element subset $\mathcal{C}'$ of $\mathcal{C}$ that is a solution of X3C if and only 2-BMG Editing/2-BMG EBEG with input $(\vec{G}, \sigma, k)$ has a yes-answer. We emphasize that the coloring σ remains unchanged in the proof below.

First suppose that X3C with input $\mathfrak{S}$ and $\mathcal{C}$ has a yes-answer. Thus, there is a t-element subset $\mathcal{C}'$ of $\mathcal{C}$ such that $\bigcup_{C \in \mathcal{C}'} C = \mathfrak{S}$. We construct a set F and add, for all $C_i \in \mathcal{C} \setminus \mathcal{C}'$ and all $s \in C_i$, the arcs (x, s_w) for every black vertex $x \in X_i$ and the arcs (x, s_b) for every white vertex $x \in X_i$. Since $|C_i| = 3$ for every $C_i \in \mathcal{C}$ and $|\mathcal{C} \setminus \mathcal{C}'| = m - t$, the set F contains exactly $6r(m-t)$ arcs, so far. Now, we add to F all arcs (s_b, s'_w) and (s_w, s'_b) whenever the corresponding elements s and s' belong to distinct elements in $\mathcal{C}'$, i.e., there is no $C \in \mathcal{C}'$ with $\{s, s'\} \subset C$. Therefore, the subgraph of $\vec{G} - F$ induced by $\mathfrak{S}$ is the disjoint union of t bi-cliques, each consisting of exactly 3 black vertices, 3 white vertices, and 18 arcs. Hence, F contains, in addition to the $6r(m-t)$ arcs, further $r - 18t$ arcs. Thus $|F| = k$. This completes the construction of F.

Since F contains only arcs but no non-arcs of $\vec{G}$, we have $\vec{G} \triangle F = \vec{G} - F$. It remains to show that $\vec{G} \triangle F$ is a (binary-explainable) BMG. To this end observe that $\vec{G} \triangle F$ has precisely m connected components that are either induced by $X_i \cup Y_i$ (in case $C_i \in \mathcal{C} \setminus \mathcal{C}'$) or $X_i \cup Y_i \cup S'$ where S' is a bi-clique containing the six vertices corresponding to the elements in $C_i \in \mathcal{C}'$. In particular, each of these components corresponds to the subgraph as specified in Lemma 5.1. To see this, note that the arcs in each connected component are given by (i) all arcs from the black to the white vertices in $X := X_i$, (ii) all arcs (x, y) with $x \in X$ and $y \in Y_i$ (or $y \in Y_i \cup S'$, respectively), and (iii) all arcs (y_1, y_2) such that y_1 and y_2 are both contained in Y_i (or in the same set in $\{Y_i, S'\}$, respectively). In particular, Lemma 5.1 implies that the disjoint union, i.e. $(\vec{G} \triangle F, \sigma)$, is a binary-explainable BMG. Hence, both BMG Editing and 2-BMG EBEG with input $(\vec{G}, \sigma, k)$ have a yes-answer.

Now, suppose that 2-BMG Editing/2-BMG EBEG with input $(\vec{G}, \sigma)$ has a yes-answer. Thus, there is a set F with $|F| \leq k$ such that $(\vec{G} \triangle F, \sigma)$ is a BMG. We will prove that we have to delete an arc set similar to the one as constructed above. First note that the number of vertices affected by F, i.e. vertices incident to inserted/deleted arcs, is at most $2k$. Since $2k < q = |\{y \in Y_i \mid \sigma(y) = \text{black}\}| = |\{y \in Y_i \mid \sigma(y) = \text{white}\}|$ for every $1 \leq i \leq m$, we have at least on black vertex $b_i \in Y_i$ and at least one white vertex $w_i \in Y_i$ that are unaffected by F. Recall that S is the bi-clique that we have constructed from a black vertex s_b and a white vertex s_w for every $s \in \mathfrak{S}$. We continue by proving

Claim 5.2.1. *Every vertex $s \in S$ has in-arcs from at most one X_i in $\vec{G} \triangle F$.*

> *Proof of Claim:* Assume w.l.o.g. that s is black and, for contradiction, that there are two distinct vertices $x_1 \in X_i$ and $x_2 \in X_j$ with $i \neq j$ and $(x_1, s), (x_2, s) \in E \triangle F$. Clearly, both x_1 and x_2 are white. As argued above, there are two (distinct) black vertices $b_1 \in Y_i$ and $b_2 \in Y_j$ that are not affected by F. Thus, (x_1, b_1) and (x_2, b_2) remain arcs in $\vec{G} \triangle F$, whereas (x_1, b_2) and (x_2, b_1) are not arcs in $\vec{G} \triangle F$, since they do not form arcs in $\vec{G}$. In summary, we have five distinct vertices x_1, x_2, b_1, b_2, s with $\sigma(x_1) = \sigma(x_2) \neq \sigma(b_1) = \sigma(b_2) = \sigma(s)$, arcs $(x_1, b_1), (x_2, b_2), (x_1, s), (x_2, s)$ and non-arcs $(x_1, b_2), (x_2, b_1)$. Thus $(\vec{G} \triangle F, \sigma)$ contains an induced F3-graph. By Lemma 4.24, $(\vec{G} \triangle F, \sigma)$ is not a BMG; a contradiction. ◇

By Claim 5.2.1, every vertex in S has in-arcs from at most one X_i. Note each X_i has r black and r white vertices. Since each element in S is either white or black, each single element in S has at most r in-arcs. Since $|S| = 2n$, we obtain at most $2rn = 2r(3t) = 6rt$ such arcs in $\vec{G} \triangle F$. In $\vec{G}$, there are in total $6rm$ arcs from the vertices in all X_i to the vertices in S. By Claim 5.2.1, F contains at least $6r(m - t)$ deletions. It remains to specify the other at most $r - 18t$ arc modifications. To this end, we show first

Claim 5.2.2. *Every vertex $s \in S$ has in-arcs from precisely one X_i in $\vec{G} \triangle F$.*

> *Proof of Claim:* Assume that there is a vertex $s \in S$ that has no in-arc from any X_i. Hence, to the aforementioned $6r(m - t)$ deletions we must add r further deletions. However, at most $r - 18t$ further edits are allowed; a contradiction. ◇

So far, F contains only arc-deletions. For the next arguments, we need the following two statements:

Claim 5.2.3. *The modification set F does not insert any arcs between X_i and X_j with $i \neq j$.*

> *Proof of Claim:* Assume for contradiction that F, and thus $\vec{G} \triangle F$, contains an arc (x_1, x_2) with $x_1 \in X_i$, $x_2 \in X_j$ and $i \neq j$. W.l.o.g. assume that x_1 is white and x_2 is black. As argued above there are black, resp., white vertices $b, w \in Y_j$ that are unaffected by F. Therefore, (x_2, w) and (b, w) remain arcs in $\vec{G} \triangle F$, whereas (x_1, b) and (b, x_1) are not arcs in $\vec{G} \triangle F$ since they do not form arcs in $\vec{G}$. In summary, $(x_1, x_2), (b, w), (x_2, w)$ are arcs in $\vec{G} \triangle F$ while $(x_1, b), (b, x_1)$ are not arcs in $\vec{G} \triangle F$. Since moreover $\sigma(x_1) = \sigma(w) \neq \sigma(b) = \sigma(x_2)$, $(\vec{G} \triangle F, \sigma)$ contains an induced F1-graph. By Lemma 4.24, $(\vec{G} \triangle F, \sigma)$ is not a BMG; a contradiction. ◇

Claim 5.2.4. *Let $s_1, s_2 \in S$ be vertices with in-arcs (x_1, s_1), resp., (x_2, s_2) in $\vec{G} \triangle F$ for some $x_1 \in X_i$ and $x_2 \in X_j$ with $i \neq j$. Then (s_1, s_2) and (s_2, s_1) cannot be arcs in $\vec{G} \triangle F$.*

> *Proof of Claim:* Assume w.l.o.g. that (s_1, s_2) is an arc in $\vec{G} \triangle F$ and that s_1 is black. It follows that x_1 and s_2 are white and x_2 is black. By construction of $\vec{G}$ and by Claim 5.2.3, we clearly have $(x_1, x_2), (x_2, x_1) \notin E \triangle F$. In summary, we have four distinct vertices x_1, x_2, s_1, s_2 with $\sigma(x_1) = \sigma(s_2) \neq \sigma(s_1) = \sigma(x_2)$, arcs

$(x_1,s_1),(x_2,s_2),(s_1,s_2)$ and non-arcs $(x_1,x_2),(x_2,x_1)$ in $\vec{G} \triangle F$. Thus $(\vec{G} \triangle F,\sigma)$ contains an induced F1-graph. By Lemma 4.24, $(\vec{G} \triangle F,\sigma)$ is not a BMG; a contradiction. $\diamond$

In summary, $\vec{G} \triangle F$ has the following property: Every $s \in S$ has in-arcs from exactly one X_i, and there are no arcs between two distinct vertices s_1 and s_2 in S that have in-arcs from two different sets X_i and X_j, respectively. Since $|C_i| = 3$ for every $C_i \in \mathcal{C}$, $(\vec{G} \triangle F)[S]$ contains connected components of size at most 6, i.e., the black and white vertex for each of the three elements in C_i. Hence, the maximum number of arcs in $(\vec{G} \triangle F)[S]$ is obtained when each of its connected components contains exactly these 6 vertices and they form a bi-clique. In this case, $(\vec{G} \triangle F)[S]$ contains $18t$ arcs. We conclude that F contains at least another $r - 18t$ deletion arcs for S. Together with the at least $6r(m-t)$ deletions between the X_i and the elements of S, we have at least $6r(m-t) + r - 18t = k \geq |F|$ arc-deletions in F. Since $|F| \leq k$ by assumption, we obtain $|F| = k$.

As argued above, the subgraph induced by S is a disjoint union of t bi-cliques of 3 white and 3 black vertices each. Since all vertices of such a bi-clique have in-arcs from the same X_i and these in-arcs are also in $\vec{G}$, we readily obtain the desired partition $\mathcal{C}' \subset \mathcal{C}$ of $\mathfrak{S}$. In other words, the C_i corresponding to the X_i having out-arcs to vertices in S in the edited digraph $\vec{G} \triangle F$ induce an exact cover of $\mathfrak{S}$. $\square$

The set F constructed in the proof of Thm. 5.2 contains only arc deletions. This immediately implies

Corollary 5.1. 2-BMG Deletion *and* 2-BMG DBEG *are NP-complete.*

In order to tackle the complexity of the 2-BMG Completion and 2-BMG CBEG problem, we follow a different approach and employ a reduction from the Chain Graph Completion problem. To this end, we need some additional notation. An undirected graph U is bipartite if its vertex set can be partitioned into two non-empty disjoint sets P and Q such that $V(U) = P \mathbin{\dot\cup} Q$ and every edge has one endpoint in P and the other endpoint in Q. In the following, we write $U = (P \mathbin{\dot\cup} Q, \widetilde{E})$ to emphasize that $\widetilde{E}$ is a set of (undirected) edges and that U is bipartite. Thus U is bipartite if and only if $x \in P$ implies $N(x) \subseteq Q$ and $x \in Q$ implies $N(x) \subseteq P$.

Definition 5.2. [cf. 220, 337] *An undirected, bipartite graph* $U = (P \mathbin{\dot\cup} Q, \widetilde{E})$ *is a* chain graph *if there is an order* $\lessdot$ *on* P *such that* $u \lessdot v$ *implies* $N(u) \subseteq N(v)$.

The Chain Graph Completion problem consists of finding a minimum-sized set of additional edges that converts an arbitrary undirected, bipartite graph into a chain graph. More formally, its decision version can be stated as follows:

Problem 5.6 (Chain Graph Completion (CGC))**.**

Input: *An undirected, bipartite graph* $U = (P \mathbin{\dot\cup} Q, \widetilde{E})$ *and an integer* k.

Question: *Is there a subset* $\widetilde{F} \subseteq \{pq \mid (p,q) \in P \times Q\} \setminus \widetilde{E}$ *such that* $|\widetilde{F}| \leq k$ *and* $U' := (P \mathbin{\dot\cup} Q, \widetilde{E} \cup \widetilde{F})$ *is a chain graph?*

It is shown in [337] that CGC is NP-complete. Following [337], we say that two edges uv and xy in an undirected graph U are *independent* if u,v,x,y are pairwise distinct and the subgraph $U[\{u,v,x,y\}]$ contains no additional edges. We will need the following characterization of chain graphs:

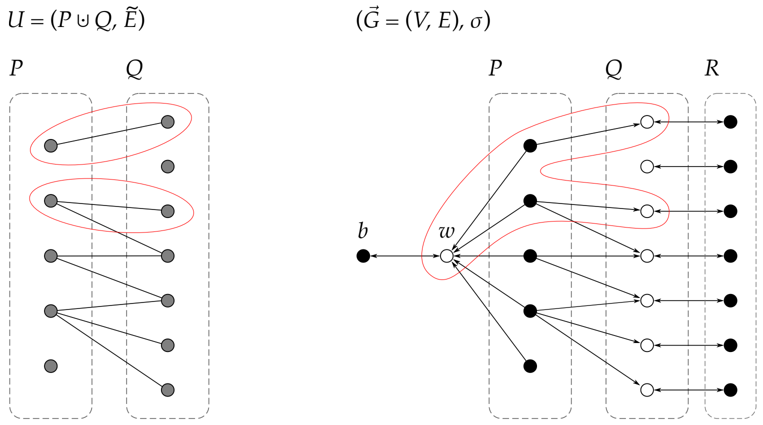

$U = (P \mathbin{\dot\cup} Q, \widetilde{E})$
$(\vec{G} = (V, E), \sigma)$
P
Q
P
Q
R
b
w

Fig. 27. Illustration of the reduction from CGC. A pair of independent edges in U and the corresponding induced F3-graph in $(\vec{G}, \sigma)$ are highlighted.

Lemma 5.2. [337, Lemma 1] *An undirected, bipartite graph* $U = (P \mathbin{\dot\cup} Q, \widetilde{E})$ *is a chain graph if and only if it does not contain a pair of independent edges.*

Theorem 5.3. 2-BMG Completion *is NP-complete.*

Proof. Since BMGs can be recognized in polynomial time by Cor. 4.6, 2-BMG Completion is clearly contained in NP. To show NP-hardness, we use a reduction from CGC. Let $(U = (P \mathbin{\dot\cup} Q, \widetilde{E}), k)$ be an instance of CGC with vertex sets $P = \{p_1, \dots, p_{|P|}\}$ and $Q = \{q_1, \dots, q_{|Q|}\}$. To construct an instance $(\vec{G} = (V, E), \sigma, k)$ of the 2-BMG Completion problem, we set $V = P \mathbin{\dot\cup} Q \mathbin{\dot\cup} R \mathbin{\dot\cup} \{b\} \mathbin{\dot\cup} \{w\}$ where $R = \{r_1, \dots, r_{|Q|}\}$ is a copy of Q. The vertices are colored $\sigma(p_i) = \sigma(r_j) = \sigma(b) =$ black and $\sigma(q_i) = \sigma(w) =$ white. The arc set E contains (q_i, r_i) and (r_i, q_i) for $1 \leq i \leq |Q|$, (p_i, w) for $1 \leq i \leq |P|$, (w, b) and (b, w), and (p, q) for every $pq \in \widetilde{E}$. This construction is illustrated in Fig. 27. Clearly, $(\vec{G}, \sigma)$ is properly colored, and the reduction can be computed in polynomial time. Moreover, it is easy to verify that $(\vec{G}, \sigma)$ is sink-free by construction, and thus, any digraph $(\vec{G}', \sigma)$ obtained from $(\vec{G}, \sigma)$ by adding arcs is also sink-free. As above, we emphasize that the coloring σ remains unchanged in the completion process.

A pair $(F, \widetilde{F})$ with $F \subseteq P \times Q$ and an edge set $\widetilde{F} = \{pq \mid (p, q) \in F\}$ will be called a *completion pair* for the bipartite graph $U = (P \mathbin{\dot\cup} Q, \widetilde{E})$ and the corresponding 2-colored digraph $(\vec{G} = (V, E), \sigma)$.

Claim 5.3.1. *If* $(F, \widetilde{F})$ *is a completion pair, then* $|F| = |\widetilde{F}|$, $(p, q) \in F$ *if and only if* $pq \in \widetilde{F}$, *and* $(p, q) \in F \cup E$ *if and only if* $pq \in \widetilde{F} \cup \widetilde{E}$.

> *Proof of Claim:* First note that, by construction, F contains only arcs from vertices in P to vertices in Q. This together with the definition $\widetilde{F} = \{pq \mid (p, q) \in F\}$ clearly implies $(p, q) \in F$ if and only if $pq \in \widetilde{F}$ and thus $|F| = |\widetilde{F}|$. By construction of our reduction we have $(p, q) \in E$ if and only if $pq \in \widetilde{E}$ and thus also $(p, q) \in E \cup F$ if and only if $pq \in \widetilde{E} \cup \widetilde{F}$. ◇

Before we continue, observe that, for every pair of independent edges $p_1q_1, p_2q_2 \in \widetilde{E}$, the subgraph of $(\vec{G}, \sigma)$ induced by $\{p_1, p_2, q_1, q_2, w\}$ is an F3-graph. Together with Lemmas 4.24 and 5.2, this implies that $(\vec{G}, \sigma)$ cannot be a BMG if U is not a chain graph. Eliminating these induced F3-graphs is closely connected to chain graph completion. More precisely we will show:

Claim 5.3.2. *Let* $(F, \widetilde{F})$ *be a completion pair. If* $(\vec{G} + F, \sigma)$ *is a BMG, then* $U' = (P \mathbin{\dot\cup} Q, \widetilde{E} \cup \widetilde{F})$ *is a chain graph.*

Proof of Claim: Suppose that $(\vec{G}+F,\sigma)$ is a BMG and assume, for contradiction, that $U' = (P \mathbin{\dot\cup} Q, \widetilde{E} \cup \widetilde{F})$ is not a chain graph. The latter and Lemma 5.2 imply that U' has two independent edges $p_1q_1, p_2q_2 \in \widetilde{E} \cup \widetilde{F}$. Thus $p_1q_2, p_2q_1 \notin \widetilde{E} \cup \widetilde{F}$. The latter arguments and Claim 5.3.1 imply that $(p_1,q_1),(p_2,q_2) \in E \cup F$ and $(p_1,q_2),(p_2,q_1) \notin E \cup F$. Since moreover $(p_1,w),(p_2,w)$ and $\sigma(p_1) = \sigma(p_2) \neq \sigma(q_1) = \sigma(q_2) = \sigma(w)$, it follows that the five distinct vertices p_1,p_2,q_1,q_2,w induce an F3-graph in $(\vec{G}+F,\sigma)$. By Lemma 4.24, $(\vec{G}+F,\sigma)$ cannot be a BMG; a contradiction. ◇

The converse is also true:

Claim 5.3.3. *Let $(F,\widetilde{F})$ be a completion pair for $U = (P \mathbin{\dot\cup} Q, \widetilde{E})$, and suppose $U' = (P \mathbin{\dot\cup} Q, \widetilde{E} \cup \widetilde{F})$ is a chain graph. Then $(\vec{G}+F,\sigma)$ is a BMG.*

Proof of Claim: By Thm. 4.7, $(\vec{G}+F,\sigma)$ is a 2-colored BMG if and and only if it is sink-free and does not contain an induced F1-, F2-, or F3-graph. Since $(\vec{G},\sigma)$ is sink-free, this is also true for $(\vec{G}+F,\sigma)$. Thus it suffices to show that $(\vec{G}+F,\sigma)$ does not contain an induced F1-, F2-, or F3-graph.

Suppose that $(\vec{G}+F,\sigma)[u,u',v,v']$ is an induced F1-graph. Let H be a subgraph of $(\vec{G}+F,\sigma)[u,u',v,v']$ that is isomorphic to the *essential* F1-graph, that is, the F1-graph as specified in Fig. 19 that contains only the solid-lined arcs and none of the dashed arcs while all other non-arcs remain non-arcs. In this case, there is an isomorphism φ from H to the essential F1-graph with vertex-labeling as in Fig. 19. Hence, $\varphi(u)$ corresponds to one of the vertices x_1, x_2, y_1 or y_2. To simplify the presentation we will say that, in this case, "u plays the role of $\varphi(u)$ in an F1-graph".

The latter definition naturally extends to $F2$- and $F3$-graphs and we will use analogous language for $F2$- and $F3$-graphs. Note, in the latter definition, it is not required that $\sigma(u) = \sigma(\varphi(u))$. Nevertheless, for $a,b \in \{u,u',v,v'\}$ with $\sigma(a) \neq \sigma(b)$ it always holds, by construction, that $\sigma(\varphi(a)) \neq \sigma(\varphi(b))$.

In the following, an in- or out-neighbor of a vertex is just called *neighbor*. A *flank vertex* in an F1-, F2-, resp., F3-graph is a vertex that has only a single neighbor in the essential F1-, F2-, resp., F3-graph. To be more precise, when referring to Fig. 19, the flank vertices in an F1-graph and F2-graph are x_1 and y_2, while the flank vertices in an F3-graph are y_1 and y_2.

Since $(F,\widetilde{F})$ is a completion pair, by definition, F adds only arcs from P to Q. Hence, each of the vertices in $R \cup \{b\}$ has a single neighbor in $(\vec{G}+F,\sigma)$ irrespective of the choice of F. Therefore, if $u \in R \cup \{b\}$ is contained in an induced F1-, F2-, or F3-graph in $(\vec{G},\sigma)$ or $(\vec{G}+F,\sigma)$, it must be a flank vertex. Observe first that b can only play the role of y_2 in the F1- or F2-graph, since otherwise, the fact that w is the single neighbor of b in $(\vec{G},\sigma)$ or $(\vec{G}+F,\sigma)$ implies that w must play the role of y_1 in the F1- or F2-graph, which is not possible since b is the single out-neighbor of w and F does not affect w. By similar arguments, none of the vertices in $R \cup \{b\}$ can play the role of x_1 in an F1- or F2-graph, or the role of y_1 or y_2 in an F3-graph in $(\vec{G},\sigma)$ or $(\vec{G}+F,\sigma)$. The vertex w has only in-arcs from the elements in P and from b. Likewise, the vertices $q_i \in Q$ have only in-arcs from P and from their corresponding vertex $r_i \in R$. Therefore and since all elements in P have only out-neighbors, it is an easy task to verify that none of the vertices in $R \cup \{b\}$ can play the role of y_2 in an F1- or F2-graph. Thus none of the vertices in $R \cup \{b\}$ is part of an induced F1-, F2-, or F3-graph.

Thus it suffices to investigate the subgraph $(\vec{G}',\sigma)$ of $(\vec{G}+F,\sigma)$ induced by $\{w\} \cup P \cup Q$ for the presence of induced F1-, F2-, and F3-graphs. In $\vec{G}'$, none of the vertices in $\{w\} \cup Q$ have out-neighbors since $F \subseteq P \times Q$ does not affect w and does not contain arcs from $q_i \in Q$ to any other vertex. Thus, none of the vertices in

$\{w\} \cup Q$ can play the role of x_1, y_1 or y_2 in an F1-, the role of x_1, y_1 or x_2 in an F2-graph, or the role of x_1 or x_2 in an F3-graph. Since $\{w\} \cup Q$ has only in-arcs from P, and P has no in-arcs in $\vec{G}'$, none of the vertices in $\{w\} \cup Q$ can play the role of x_2 in an F1-graphs or the role of y_2 in an F2-graph. Thus none of the vertices in $\{w\} \cup Q$ is part of an induced F1- or F2-graph. Hence, any induced F1- or F2-graph must be contained in $\vec{G}'[P]$. However, all vertices of P are colored black, and hence $(\vec{G}'[P], \sigma_{|P})$ cannot harbor an induced F1- or F2-graph.

Suppose $(\vec{G}', \sigma)$ contains an induced F3-graph. Then there are five pairwise distinct vertices $x_1, x_2, y_1, y_2, y_3 \in \{w\} \cup P \cup Q$ with coloring $\sigma(x_1) = \sigma(x_2) \neq \sigma(y_1) = \sigma(y_2) = \sigma(y_3)$ satisfying $(x_1, y_1), (x_2, y_2), (x_1, y_3), (x_2, y_3) \in E \cup F$ and $(x_1, y_2), (x_2, y_1) \notin E \cup F$. Since P has no in-arcs in $(\vec{G}', \sigma)$, it must hold that $y_1, y_2, y_3 \notin P$. Since $\sigma(\{w\} \cup Q) \neq \sigma(P)$ and $(\vec{G}', \sigma)$ is properly 2-colored, we have $x_1, x_2 \in P$. Since w has in-arcs from all vertices in P and $(x_1, y_2), (x_2, y_1) \notin E \cup F$, vertex w can neither play the role of y_1 nor of y_2 in an F3-subgraph. Thus, $y_1, y_2 \in Q$. Claim 5.3.1 therefore implies $x_1y_1, x_2y_2 \in \widetilde{E} \cup \widetilde{F}$ and $x_1y_2, x_2y_1 \notin \widetilde{E} \cup \widetilde{F}$. Hence, U' contains a pair of independent edges. By Lemma 5.2, it follows that U' is not a chain graph; a contradiction. ◇

Together, Claims 5.3.2 and 5.3.3 imply that $(\vec{G} + F, \sigma)$ is a BMG if and only if $U' = (P \,\dot\cup\, Q, \widetilde{E} \cup \widetilde{F})$ is a chain graph; see Fig. 28 for an illustrative example.

Claim 5.3.4. *If F is a minimum-sized arc completion set such that $(\vec{G} + F, \sigma)$ is a BMG, then $F \subseteq P \times Q$.*

Proof of Claim: Let F be an arbitrary minimum-sized arc completion set, i.e., $(\vec{G} + F, \sigma)$ is a BMG, and put $F' := F \cap (P \times Q)$ and let $(F', \widetilde{F}')$ be the corresponding completion pair.

If $F' = F$, there is nothing to show. Otherwise, we have $|F'| < |F|$ and minimality of $|F|$ implies that $(\vec{G} + F', \sigma)$ is not a BMG. By contraposition of Claim 5.3.3, we infer that $U' = (P \,\dot\cup\, Q, \widetilde{E} \cup \widetilde{F}')$ is not a chain graph. Hence, Lemma 5.2 implies that U' contains a set of independent edges $p_1q_1, p_2q_2 \in \widetilde{E} \cup \widetilde{F}'$ and $p_1q_2, p_2q_1 \notin \widetilde{E} \cup \widetilde{F}'$. By Claim 5.3.1, it follows that $(p_1, q_1), (p_2, q_2) \in E \cup F'$ and $(p_1, q_2), (p_2, q_1) \notin E \cup F'$. Since $F' \subset F$, we have $(p_1, q_1), (p_2, q_2) \in E \cup F$. Furthermore, from $(p_1, q_2), (p_2, q_1) \in P \times Q$ and $F' = F \cap (P \times Q)$, we conclude that $(p_1, q_2), (p_2, q_1) \notin E \cup F$. By construction of our reduction and since we only insert arcs, we have $(p_1, w), (p_2, w) \in E \cup F$. Together with the coloring $\sigma(p_1) = \sigma(p_2) \neq \sigma(q_1) = \sigma(q_2) = \sigma(w)$, the latter arguments imply that $(\vec{G} + F, \sigma)$ contains an induced F3-graph. By Lemma 4.24, this contradicts that $(\vec{G} + F, \sigma)$ is a BMG. ◇

Now, let $(F, \widetilde{F})$ be a completion pair such that $|\widetilde{F}| \leq k$ and $\widetilde{F}$ is a minimum-sized edge completion set for U. Thus $U' = (P \,\dot\cup\, Q, \widetilde{E} \cup \widetilde{F})$ is a chain graph. Hence, Claim 5.3.3 implies that $(\vec{G} + F, \sigma)$ is a BMG. Since $|F| = |\widetilde{F}| \leq k$, it follows that 2-BMG Completion with input $(\vec{G}, \sigma, k)$ has a yes-answer if CGC with input $(U = (P \,\dot\cup\, Q, \widetilde{E}), k)$ has a yes-answer.

Finally, let F be a minimum-sized arc completion set for $(\vec{G}, \sigma)$, i.e. $(\vec{G} + F, \sigma)$ is a BMG, and assume $|F| \leq k$. This and Claim 5.3.4 implies $F \subseteq P \times Q$. For the corresponding completion pair $(F, \widetilde{F})$ we have $|\widetilde{F}| = |F| \leq k$. Moreover, since $(\vec{G} + F, \sigma)$ is a BMG, Claim 5.3.2 implies that $U = (P \,\dot\cup\, Q, \widetilde{E} \cup \widetilde{F})$ is a chain graph. Therefore, CGC with input $(U = (P \,\dot\cup\, Q, \widetilde{E}), k)$ has a yes-answer if 2-BMG Completion with input $(\vec{G}, \sigma, k)$ has a yes-answer. This completes the proof. □

The reduction employed in the proof of Thm. 5.3 can be adapted to show that the 2-BMG CBEG problem is hard. In particular, neither the original

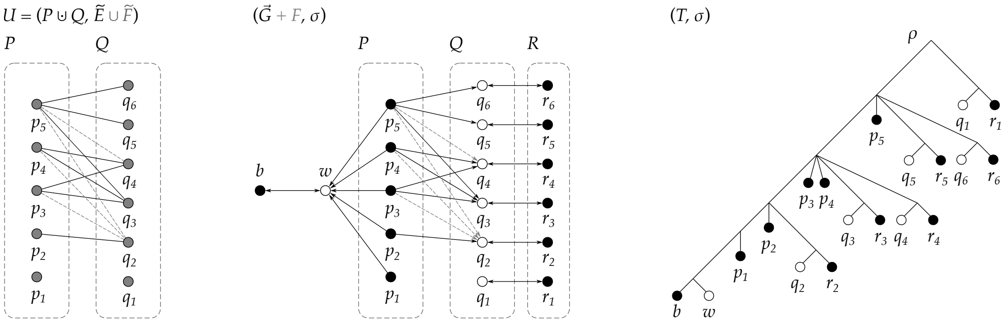


Fig. 28. An example solution for CGC, resp., 2-BMG Completion as constructed in the proof of Thm. 5.3. A tree (T, σ) that explains the resulting BMG is shown on the right. Here, we have $k = 4$ edge, resp., arc additions (indicated by dashed-gray lines) to obtain a chain graph, resp., 2-BMG. The indices of the vertices in $P = \{p_1, \dots, p_{|P|}\}$ are chosen w.r.t. the order $\lessdot$ on P i.e. $i < j$ if and only if $p_i \lessdot p_j$ and thus, $N(p_i) \subseteq N(p_j)$. In this example, we have $N(p_1) \cap Q = \emptyset$. Moreover, the vertex q_1 has no neighbor in P.

graph construction in the reduction nor the (optimally) modified version contain hourglasses. Hence, we have

Corollary 5.2. 2-BMG CBEG *is NP-complete.*

Proof. By Cor. 4.10, binary-explainable BMGs can be recognized in polynomial time. Therefore, 2-BMG CBEG is contained in the class NP.

To show hardness of the problem, we use the same reduction from CGC and the same arguments as in the proof of Thm. 5.3. In addition, we observe that the hourglass $[xy \bowtie x'y']$ contains the bidirectional arcs (x, y) and (y, x) and each of the two vertices x and y has two out-neighbors, and thus, also at least two out-neighbors in every digraph that contains the hourglass as an induced subgraph.

We have to show that CGC with input $(U = (P \cup Q, \widetilde{E}), k)$ has a yes-answer if and only if 2-BMG CBEG with input $(\vec{G}, \sigma, k)$ as constructed in the proof of Thm. 5.3 has a yes-answer. Recall that by Claims 5.3.2 and 5.3.3, $(\vec{G} + F, \sigma)$ is a BMG if and only if $U' = (P \cup Q, \widetilde{E} \cup \widetilde{F})$ is a chain graph, where $(F, \widetilde{F})$ is a completion pair. Moreover, by Claim 5.3.4, every minimum-sized arc completion set F for which $(\vec{G} + F, \sigma)$ is a BMG satisfies $F \subseteq P \times Q$. Therefore, we can again argue via minimal completion pairs $(F, \widetilde{F})$ to conclude that, both in the *if*- and in the *only-if*-direction, we have a 2-BMG $(\vec{G} + F, \sigma)$ with $F \subseteq P \times Q$, i.e., we only inserted arcs from P to Q. Using Fig. 28, it is now easy to verify that every bidirectional pair of arcs in $(\vec{G} + F, \sigma)$ is either incident to the vertex b or to one of the vertices in R. Moreover, every vertex in $R \cup \{b\}$ has exactly one out-neighbor. The latter two arguments together with the observation that hourglasses require bidirectional arcs $(x, y), (y, x)$ such that both x and y have at least two out-neighbors imply that $(\vec{G} + F, \sigma)$ must be hourglass-free. Therefore, $(\vec{G} + F, \sigma)$ is binary-explainable by Prop. 4.8, which completes the proof. ☐

In summary, all three variants of 2-BMG modification problems as well as their beBMG counterparts are NP-complete.

However, we will now turn to an interesting restriction of the 2-BMG CBEG problem, namely the completion of a 2-colored digraph that is already a BMG to a binary-explainable BMG, which we will show to have a unique optimal solution that can be constructed in polynomial time. To this end, we start with the following result which holds for BMGs and their completions to beBMGs with an arbitrary number of colors.

Lemma 5.3. *Let* $(\vec{G}', \sigma)$ *be a completion of a BMG* $(\vec{G}, \sigma)$ *to a beBMG, and let* $[xy \mathrel{\rlap{$\nearrow$}\searrow} x'y']$ *be an induced hourglass in* $(\vec{G}, \sigma)$*. Then* $(\vec{G}', \sigma)$ *contains both arcs* (x', y) *and* (y', x)*.*

Proof. As a direct consequence of Obs. 4.2, the subgraphs of a BMG induced by all vertices with any two given colors is a 2-BMG. Since $(\vec{G}', \sigma)$ is a (binary-explainable) BMG, all of its 2-colored induced subgraphs are therefore 2-BMGs. By assumption, $(\vec{G}, \sigma)$ is not binary-explainable since it contains the hourglass $[xy \mathrel{\rlap{$\nearrow$}\searrow} x'y']$ as an induced subgraph (cf. Prop. 4.8). The hourglass contains all possible arcs between vertices of different colors except (x', y) and (y', x). Since $(\vec{G}', \sigma)$ contains no hourglass, and $\vec{G}'$ is a completion of $\vec{G}$, i.e., $E(\vec{G}) \subseteq E(\vec{G}')$, we conclude that $(\vec{G}', \sigma)$ contains at least one of the arcs (x', y) and (y', x).

Assume for contradiction that, w.l.o.g., $(\vec{G}', \sigma)$ only contains (x', y). We have $(y', x'), (y, x) \in E(\vec{G}')$ and $\sigma(y') = \sigma(y) \neq \sigma(x') = \sigma(x)$ by the definition of hourglasses, and by assumption $(x', y) \in E(\vec{G}')$ and $(y', x) \notin E(\vec{G}')$. Hence, the four vertices x, x', y, y' induce an F2-graph in $(\vec{G}', \sigma)$. By Thm. 4.7, the 2-colored subgraph of $(\vec{G}', \sigma)$ induced by the two colors $\sigma(x)$ and $\sigma(y)$ is not a BMG. Consequently, $(\vec{G}', \sigma)$ is not a BMG either; a contradiction. Hence, $(\vec{G}', \sigma)$ contains both arcs (x', y) and (y', x). ☐

For the following, recall the definition of support leaves (Def. 4.15) and that every inner vertex u of the LRT of a 2-BMG $(\vec{G}, \sigma)$, with the possible exception of the root ρ, has a non-empty set of support leaves S_u, and $S_\rho \neq \emptyset$ if and only if $(\vec{G}, \sigma)$ is connected (Lemma 4.28).

Definition 5.3. *Let* (T, σ) *be a tree with a 2-colored leaf set, i.e.,* $|\sigma(L(T))| = 2$*. Denote by* (T^c, σ) *the* collapsed tree *obtained from* (T, σ) *by contraction of all inner edges in* $T(u)$ *for all* $u \in V^0(T)$ *that have support leaves of both colors.*

In other words, the collapsed tree (T^c, σ) is obtained from (T, σ) by collapsing every subtree $T(u)$ to a star if u has support leaves of both colors.

Lemma 5.4. *The collapsed tree* (T^c, σ) *of* (T, σ) *is uniquely defined and can be computed from* (T, σ) *in* $O(|V(T)|)$*-time.*

Proof. The collapsed tree (T^c, σ) is well-defined because whenever $v \prec_T u$, then collapsing the subtree $T(v)$ to a star does not change the set of support leaves S_u. Similarly, collapsing $T(v)$ if v is not $\prec_T$-comparable with u does not change S_u. Thus (T^c, σ) is uniquely defined. To see that (T^c, σ) can be computed in $O(|V(T)|)$ operations, we observe that it suffices to collapse all subtrees $T(u)$ such that $u \in V^0(T)$ has support leaves of both colors and there is no $u' \prec_T u$ with this property, i.e., u is $\preceq_T$-maximal in that sense. These vertices u for which $T(u)$ is replaced by a star are found by a top-down traversal of T and evaluating $|\sigma(S_u)|$, all of which can be computed in linear total time. ☐

As an immediate consequence of the uniqueness of (T^c, σ) and the construction in the second part of the proof of Lemma 5.4, we obtain

Corollary 5.3. *The collapsed tree* $((T^c)^c, \sigma)$ *of a collapsed tree* (T^c, σ) *satisfies* $(T^c)^c = T^c$*.*

Lemma 5.5. *If (T^c,σ) is the collapsed tree of an LRT (T,σ) with 2-colored leaf set, then $\vec{G}(T^c,\sigma)$ is binary-explainable.*

Proof. Since the collapsed tree (T^c,σ) is obtained from the LRT (T,σ) by contraction of edges, Prop. 4.2 implies that (T^c,σ) is also least resolved. Now suppose, for contradiction, that $\vec{G}(T^c,\sigma)$ is not binary-explainable. By, Prop. 4.9(3), (T^c,σ) has a vertex $u \in V^0(T^c)$ with three distinct children v_1, v_2, and v_3 and two distinct colors r and s satisfying (i) $r \in \sigma(L(T^c(v_1)))$, $r,s \in \sigma(L(T^c(v_2)))$, and $s \in \sigma(L(T^c(v_3)))$, and (ii) $s \notin \sigma(L(T^c(v_1)))$, and $r \notin \sigma(L(T^c(v_3)))$. Since $(\vec{G},\sigma)$ is only 2-colored, the latter arguments imply that $|\sigma(L(T^c(v_1)))| = |\sigma(L(T^c(v_3)))| = 1$ and $|\sigma(L(T^c(v_2))| = 2$. Since moreover (T^c,σ) is least resolved and none of the vertices v_1, v_2, and v_3 is the root of T^c, we can apply Lemma 4.26 to conclude that v_1 and v_2 are leaves, and that v_3 is an inner vertex, respectively. In particular, $\sigma(v_1) = r \neq s = \sigma(v_3)$. Hence, $T^c(u)$ is not a star tree and u has support leaves of both colors in T^c; a contradiction to its construction. Therefore, we conclude that $\vec{G}(T^c,\sigma)$ is binary-explainable. ☐

Theorem 5.4. *The optimization version of* 2-BMG CBEG *with a 2-BMG $(\vec{G},\sigma)$ as input has the unique solution $F := E(\vec{G}(T^c,\sigma)) \setminus E(\vec{G})$, where (T^c,σ) is the collapsed tree of the LRT (T,σ) of $(\vec{G},\sigma)$.*

Proof. First note that the optimization version of 2-BMG CBEG always has a solution. To see this, consider the complete bipartite and properly 2-colored digraph $(\vec{G}',\sigma)$ with vertex set $V(\vec{G})$. This digraph is explained by the star tree with leaf set $V(\vec{G})$. Moreover, $(\vec{G}',\sigma)$ is clearly hourglass-free since hourglasses require non-arcs (between vertices of distinct colors). By Prop. 4.8, the BMG $(\vec{G}',\sigma)$ is binary-explainable.

Now consider the collapsed tree (T^c,σ) of (T,σ). Since T^c is obtained from T by contraction of inner edges, Lemma 4.7 implies $(\vec{G},\sigma) = \vec{G}(T,\sigma) \subseteq \vec{G}(T^c,\sigma) =: (\vec{G}^*,\sigma)$. Furthermore, $(\vec{G}^*,\sigma)$ is binary-explainable by Lemma 5.5. Therefore, $(\vec{G}^*,\sigma)$ is a valid completion of $(\vec{G},\sigma)$ to a beBMG.

We continue by showing the existence of certain arcs in every (not necessarily optimal) completion $(\vec{G}',\sigma)$ of $(\vec{G},\sigma)$ to a beBMG. To this end, consider a $\preceq_T$-maximal vertex u such that the subtree $T(u)$ is not a star tree and u has support leaves S_u of both colors in T. We will make frequent use of the fact that $E(\vec{G}) \subseteq E(\vec{G}')$. We consider the following cases in order to show that all arcs between vertices $x,y \in L(T(u))$ with $\sigma(x) \neq \sigma(y)$ exist in $(\vec{G}',\sigma)$:

(i) $x,y \in S_u$,

(ii) $x \in L(T(u)) \setminus S_u$ and $y \in S_u$, and

(iii) $x,y \in L(T(u)) \setminus S_u$.

In Case (i), the leaves x and y are both children of u. Together with Cor. 4.14, this implies $(x,y),(y,x) \in E(\vec{G}) \subseteq E(\vec{G}')$.

In Case (ii), we can find a vertex $x' \in S_u$ of color $\sigma(x)$ since S_u contains vertices of both colors. As in Case (i), we have $(x',y),(y,x') \in E(\vec{G}) \subseteq E(\vec{G}')$. Since $x \in L(T(u)) \setminus S_u$, we can conclude that $v := \mathsf{par}_T(x) \prec_T u$ by the definition of support leaves. Hence, the inner vertex v is not the root of T and we can apply Cor. 4.13 to conclude that the subtree $T(v)$ of the inner vertex v contains both colors. The latter together with Lemma 4.3 implies that there are arcs $(x'',y''),(y'',x'') \in E(\vec{G}) \subseteq E(\vec{G}')$ with $x'',y'' \in L(T(v))$ and $\sigma(x) = \sigma(x'') \neq \sigma(y) = \sigma(y'')$. Note that $x = x''$ is possible. Since x,x'',y'' in $L(T(v)) \subset L(T(u))$, $x',y \in L(T(u)) \setminus L(T(v))$ and $v \prec_T u$, we can apply Lemma 4.27 to conclude that $(x',y''),(y,x),(y,x'') \in E(\vec{G}) \subseteq E(\vec{G}')$ and $(y'',x'),(x,y),(x'',y) \notin E(\vec{G}) \subseteq E(\vec{G}')$. Together with $(x',y),(y,x'),(x'',y''),(y'',x'') \in E(\vec{G})$ and the coloring, this implies that x',y,x'',y'' induce an hourglass $[x'y \bowtie x''y'']$ in $(\vec{G},\sigma)$. By Lemma 5.3, we have arcs $(x'', y), (y'', x') \in E(\vec{G}')$. If $x = x''$, we immediately obtain $(x, y), (y, x) \in E(\vec{G}')$. Now suppose $x \neq x''$, i.e., it remains to show that $(x, y) \in E(\vec{G}')$. Thus assume, for contradiction, that $(x, y) \notin E(\vec{G}')$. Lemma 4.27 together with $\sigma(x) \neq \sigma(y'')$ and $y'' \in L(T(\mathsf{par}_T(x) = v))$ implies that $(x, y'') \in E(\vec{G}) \subseteq E(\vec{G}')$. Hence, we have the arcs $(x, y''), (y'', x'), (x', y) \in E(\vec{G}')$ but $(x, y) \notin E(\vec{G}')$, i.e., x, x', y, y'' induce a forbidden F2-graph. Together with Thm. 4.7, this is a contradiction to $(\vec{G}', \sigma)$ being a 2-BMG. Therefore, we conclude that $(x, y) \in E(\vec{G}')$.

In Case (iii), we have $x, y \in L(T(u)) \setminus S_u$. We can find two vertices $x', y' \in S_u$, which are distinct from x and y and satisfy $\sigma(x) = \sigma(x') \neq \sigma(y) = \sigma(y')$. From Cases (i) and (ii), we obtain $(x', y'), (y', x') \in E(\vec{G}')$ and $(x', y), (y, x'), (x, y'), (y', x) \in E(\vec{G}')$, respectively. Now assume for contradiction that $(x, y) \notin E(\vec{G}')$. Thus, we have $(x, y'), (y', x'), (x', y) \in E(\vec{G}')$ and $(x, y) \notin E(\vec{G}')$, i.e., x, x', y, y' induce a forbidden F2-graph in $(\vec{G}', \sigma)$; a contradiction to $(\vec{G}', \sigma)$ being a 2-BMG. Hence, we conclude that $(x, y) \in E(\vec{G}')$. The existence of the arc $(y, x) \in E(\vec{G}')$ can be shown by analogous arguments.

We will now show that $E(\vec{G}^*) \subseteq E(\vec{G}')$ for every (not necessarily optimal) completion $(\vec{G}', \sigma)$ of the 2-BMG $(\vec{G}, \sigma)$ to a beBMG. To this end, consider an arbitrary arc $(x, y) \in E(\vec{G}^*)$. If $(x, y) \in E(\vec{G})$, then $(x, y) \in E(\vec{G}')$ follows immediately. Now assume that $(x, y) \in F = E(\vec{G}^*) \setminus E(\vec{G})$. Since $(\vec{G}, \sigma)$ is a 2-BMG and thus properly-colored and sink-free (cf. Thm. 4.7), there must be a vertex y' of color $\sigma(y)$ such that $(x, y') \in E(\vec{G})$. Since $(x, y) \notin E(\vec{G})$, we have $\mathrm{lca}_T(x, y') \prec_T \mathrm{lca}_T(x, y)$ and thus the LRT (T, σ) displays the triple $xy'|y$. However, $(x, y), (x, y') \in E(\vec{G}^*)$ implies that (T^c, σ) does not display the triple $xy'|y$, i.e., all edges on the path from $\mathrm{lca}_T(x, y')$ to $\mathrm{lca}_T(x, y)$ have been contracted. Therefore, there is a $\preceq_T$-maximal inner vertex $u \in V^0(T)$ such that $x, y \in L(T(u))$, $T(u)$ is not a star tree and u has support leaves of both colors in T. By the arguments above, we can conclude that $(x, y) \in E(\vec{G}')$.

In summary, F is a solution for 2-BMG CBEG with the 2-BMG $(\vec{G}, \sigma)$ (and some integer $k \geq |F|$) as input, and $F \subseteq F'$ for every other solution $F' = E(\vec{G}') \setminus E(\vec{G})$. Therefore, we conclude that F is the unique optimal solution. $\square$

As a direct consequence of Thm. 5.4, the fact that LRTs can be constructed in $O(|V| + |E| \log^2 |V|)$ (cf. Thm. 4.10) and Lemma 5.4, we have

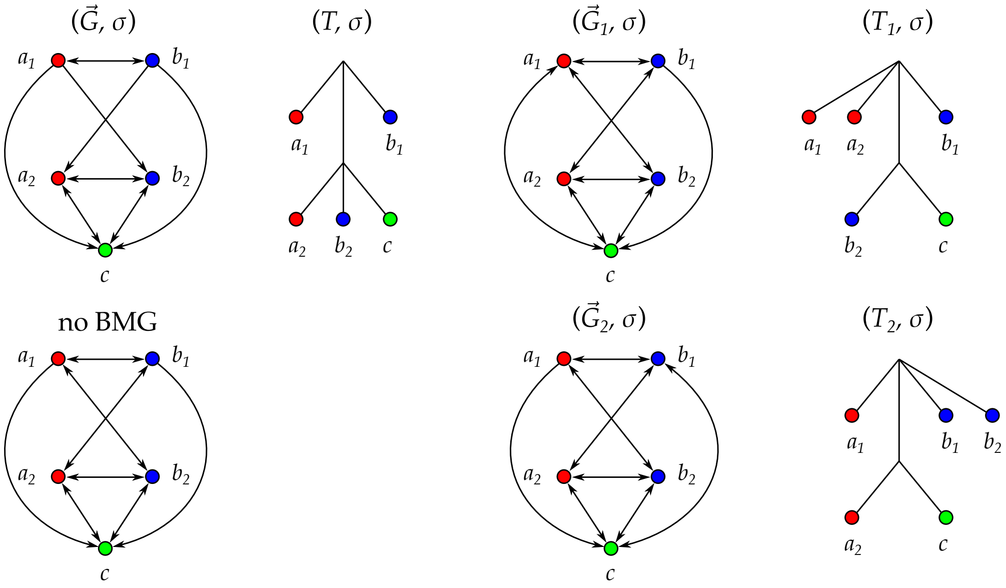
Corollary 5.4. 2-BMG CBEG *with a 2-BMG as input can be solved in* $O(|V| + |E| \log^2 |V|)$ *time.*

We also immediately obtain a characterization of the LRTs of 2-beBMGs.

Corollary 5.5. *A 2-colored least resolved tree* (T, σ) *is the LRT of 2-beBMG if and only if it is a collapsed tree.*

In contrast to the 2-colored case, ℓ-BMG CBEG with a BMG as input and $\ell \geq 3$ in general does not have a unique optimal solution. In the example in Fig. 29, the missing arcs (a_2, b_1) and (b_2, a_1) in the induced hourglass $[a_1 b_1 \text{⤨} a_2 b_2]$ must be inserted (cf. Lemma 5.3). The resulting digraph is not a BMG. To obtain a BMG, it suffices to insert in addition either the arc (c, a_1) or the arc (c, b_1) to obtain a beBMG. (cf. Prop. 4.9).

The simple solution of 2-BMG CBEG begs the question whether other arc modification problems for beBMGs, in particular the corresponding deletion and editing problems, have a similar structure. This does not seem to be case, however. Neither 2-BMG EBEG nor 2-BMG DBEG with a 2-BMG as input have a unique optimal solution. To see this, consider the 2-BMG consisting of the hourglass $[xy \text{⤨} x'y']$ which is explained by the unique non-binary tree


$(\vec{G},\sigma)$ (T,σ) $(\vec{G}_1,\sigma)$ (T_1,σ)

no BMG $(\vec{G}_2,\sigma)$ (T_2,σ)

Fig. 29. Example for 3-BMG CBEG with the 3-BMG $(\vec{G},\sigma)$ (explained by the LRT (T,σ)) as input that has no unique optimal solution. Insertion of the missing arcs (a_2,b_1) and (b_2,a_1) produces a digraph that is not a BMG. At least one of the arcs (c,a_1) or (c,b_1) has to be inserted additionally to obtain the beBMGs $(\vec{G}_1,\sigma)$ and $(\vec{G}_2,\sigma)$ (shown with their LRTs (T_1,σ) and (T_2,σ)), respectively.

$(x,y,(x',y'))$ (in Newick format, see also Fig. 13). Deletion of the arcs (x,y) or (y,x) results in a digraph that is explained by the binary trees $(y,(x,(x',y')))$ or $(x,(y,(x',y')))$, respectively. We suspect that a BMG as input does not make these problems easier than the general case – the complexity of which remains an open questions, however.

5.3 COMPLEXITY OF ℓ-BMG MODIFICATION PROBLEMS

We now turn to the graph modification problems for an arbitrary number ℓ of colors. The proof of the next theorem follows the same strategy of adding hub-vertices as in [137].

Theorem 5.5. *ℓ-BMG Deletion, ℓ-BMG Completion, and ℓ-BMG Editing are NP-complete for all $\ell \geq 2$.*

Proof. BMGs can be recognized in polynomial time by Cor. 4.6 and thus, all three problems are contained in the class NP. Let $(\vec{G}=(V,E),\sigma)$ be a properly colored digraph with ℓ colors. Thm. 5.2, Cor. 5.1 and Thm. 5.3 state NP-completeness for the case of $\ell=2$ colors. Thus assume $\ell \geq 3$ in the following.

By slight abuse of notation, we collectively refer to the three problems ℓ-BMG Deletion, ℓ-BMG Completion, and ℓ-BMG Editing simply as ℓ-BMG Modification. Correspondingly, we write $(\vec{G}\odot F,\sigma)$ and distinguish the three problems by the modification operation $\odot \in \{-,+,\triangle\}$, where $\odot=-$, $\odot=+$ and $\odot=\triangle$ specifies that F is a deletion-, completion, or edit set, respectively.

We use reduction from 2-BMG Modification. To this end, let $(\vec{G}_2=(V_2,E_2),\sigma_2,k)$ be an instance of one of the latter three problems. To obtain a properly colored digraph $(\vec{G}_\ell=(V_\ell,E_\ell),\sigma_\ell)$ with ℓ colors, we add to $\vec{G}_2$ a set V_H of $\ell-2$ new vertices with pairwise distinct colors that also do not share any colors with the vertices in

$(\vec{G}_2, \sigma_2)$. Moreover, we add arcs such that every $h \in V_H$ becomes a hub-vertex. Note that $V_\ell = V_2 \,\dot\cup\, V_H$, $\vec{G}_\ell[V_2] = \vec{G}_2$, and $(\sigma_\ell)_{|V_2} = \sigma_2$. Furthermore, V_2 is a subset of V_ℓ satisfying the condition in Obs. 4.2, i.e., $V_2 = \bigcup_{s \in S_2} V_\ell[s]$ for the color set S_2 in $(\vec{G}_2, \sigma_2)$. Clearly, the reduction can be performed in polynomial time. We proceed by showing that an instance $(\vec{G}_2, \sigma_2, k)$ of the respective 2-BMG Modification problem has a yes-answer if and only if the corresponding instance $(\vec{G}_\ell, \sigma_\ell, k)$ of ℓ-BMG Modification has a yes-answer.

Suppose that 2-BMG Modification with input $(\vec{G}_2, \sigma_2, k)$ has a yes-answer. Then there is an arc set $F \subseteq V_2 \times V_2 \setminus \{(v,v) \mid v \in V_2\}$ with $|F| \leq k$ such that $(\vec{G}_2 \odot F, \sigma_2)$ is a BMG. Let (T_2, σ_2) be a tree with root ρ explaining $(\vec{G}_2 \odot F, \sigma_2)$. Now take (T_2, σ_2) and add the vertices in V_H as leaves of the root ρ and color these leaves as in $(\vec{G}_\ell, \sigma_\ell)$, to obtain the tree (T_ℓ, σ_ℓ). By construction, we have $L(T_\ell) = V_\ell = V_2 \cup V_H$ and $T_2 = (T_\ell)_{|V_2}$, where $(T_\ell)_{|V_2}$ is the restriction of T_ℓ to the leaf set V_2. The latter arguments together with Obs. 4.2 imply that $(\vec{G}(T_\ell, \sigma_\ell)[V_2], (\sigma_\ell)_{|V_2}) = \vec{G}((T_\ell)_{|V_2}, (\sigma_\ell)_{|V_2}) = \vec{G}(T_2, \sigma_2) = (\vec{G}_2 \odot F, \sigma_2)$.

Let $h \in V_H$ be arbitrary. Since h is the only vertex of its color, (x,h) is an arc in $\vec{G}(T_\ell, \sigma_\ell)$ for every $x \in V_\ell \setminus \{h\}$. Since h is a child of the root, we have moreover $\operatorname{lca}_{T_\ell}(x,h) = \rho$, and thus, (h,x) is an arc in $\vec{G}(T_\ell, \sigma_\ell)$ for every $x \in V_\ell \setminus \{h\}$. The latter two arguments imply that h is a hub-vertex in $\vec{G}(T_\ell, \sigma_\ell)$. Since F is not incident to any vertex in $V_\ell \setminus V_2 = V_H$ and each vertex $h \in V_H$ is a hub-vertex in $(\vec{G}_\ell, \sigma_\ell)$ and in $\vec{G}(T_\ell, \sigma_\ell)$, we conclude that $\vec{G}(T_\ell, \sigma_\ell) = (\vec{G}_\ell \odot F, \sigma_\ell)$. Hence, $(\vec{G}_\ell \odot F, \sigma_\ell)$ is a BMG and the corresponding ℓ-BMG Modification problem with input $(\vec{G}_\ell, \sigma_\ell, k)$ has a yes-answer.

For the converse, suppose that ℓ-BMG Modification with input $(\vec{G}_\ell, \sigma_\ell, k)$ has a yes-answer. Thus, there is an arc set $F \subseteq V_\ell \times V_\ell \setminus \{(v,v) \mid v \in V_\ell\}$ with $|F| \leq k$ such that $(\vec{G}_\ell \odot F, \sigma_\ell)$ is a BMG. Let (T_ℓ, σ_ℓ) be a tree explaining $(\vec{G}_\ell \odot F, \sigma_\ell)$. Let $F' \subseteq F$ be the subset of arc modifications (x,y) for which $x, y \in V_2$. Thus, it holds $|F'| \leq |F| \leq k$. By construction, $(\vec{G}_\ell \odot F)[V_2] = \vec{G}_\ell[V_2] \odot F'$. Moreover, by Obs. 4.2, we have $(\vec{G}(T_\ell, \sigma_\ell)[V_2], (\sigma_\ell)_{|V_2}) = \vec{G}((T_\ell)_{|V_2}, (\sigma_\ell)_{|V_2})$. In summary, we obtain $(\vec{G}_2 \odot F', \sigma_2) = (\vec{G}_\ell[V_2] \odot F', \sigma_2) = ((\vec{G}_\ell \odot F)[V_2], (\sigma_\ell)_{|V_2}) = (\vec{G}(T_\ell, \sigma_\ell)[V_2], (\sigma_\ell)_{|V_2}) = \vec{G}((T_\ell)_{|V_2}, (\sigma_\ell)_{|V_2})$. Thus, $(\vec{G}_2 \odot F', \sigma_2)$ is a BMG. Together with $|F'| \leq k$, this implies that 2-BMG Modification with input $(\vec{G}_2, \sigma_2, k)$ has a yes-answer. ☐

As in the 2-colored case, we can reuse the reduction to show that ℓ-BMG DBEG/CBEG/EBEG are all NP-complete.

Corollary 5.6. *ℓ-BMG DBEG, ℓ-BMG CBEG and ℓ-BMG EBEG are NP-complete for all $\ell \geq 2$.*

Proof. By Cor. 4.10, binary-explainable BMGs can be recognized in polynomial time. Therefore, all three problems are contained in the class NP. Cor. 5.1, Cor. 5.2 and Thm. 5.2 state NP-completeness for the case $\ell = 2$. Thus, it remains to show NP-hardness for the case $\ell \geq 3$.

We use a reduction from 2-BMG DBEG/CBEG/EBEG and the same polynomial-time construction as in the proof of Thm. 5.5, i.e., we construct an ℓ-colored digraph $(\vec{G}_\ell = (V_\ell, E_\ell), \sigma_\ell)$ from a 2-colored digraph $(\vec{G}_2 = (V_2, E_2), \sigma_2)$ by adding a hub-vertex of $\ell - 2$ pairwise distinct new colors. As before, we write $(\vec{G} \odot F, \sigma)$ and distinguish the three problems by the modification operation $\odot \in \{-, +, \triangle\}$, where $\odot = -$, $\odot = +$ and $\odot = \triangle$ specifies that F is a deletion-, completion, or edit set, respectively. We proceed by showing that an instance $(\vec{G}_2, \sigma_2, k)$ of the respective 2-BMG DBEG/CBEG/EBEG problem has a yes-answer if and only if the corresponding instance $(\vec{G}_\ell, \sigma_\ell, k)$ of ℓ-BMG DBEG/CBEG/EBEG has a yes-answer.

First suppose that 2-BMG DBEG/CBEG/EBEG with input $(\vec{G}_2, \sigma_2, k)$ has a solution $F \subseteq V_2 \times V_2 \setminus \{(v,v) \mid v \in V_2\}$ with $|F| \leq k$ such that $(\vec{G}_2 \odot F, \sigma_2)$ is a *binary-explainable* BMG. By Prop. 4.8, $(\vec{G}_2 \odot F, \sigma_2)$ is hourglass-free. Since $(\vec{G}_2 \odot F, \sigma_2)$ is in particular a BMG, we can use the same arguments as in the proof of Thm. 5.5 to conclude that $(\vec{G}_\ell \odot F, \sigma_\ell)$ is a BMG. Now observe that an hourglass contains two vertices of each of its two colors. Therefore and since every vertex in $V_H = V_\ell \setminus V_2$ is the only vertex of its color, none the vertices in V_H is part of an induced subgraph of $(\vec{G}_\ell \odot F, \sigma_\ell)$ that is an hourglass. Hence, all hourglasses of $(\vec{G}_\ell \odot F, \sigma_\ell)$ must be part of the induced subgraph $((\vec{G}_\ell \odot F)[V_2], (\sigma_\ell)_{|V_2})$. This together with the facts that $((\vec{G}_\ell \odot F)[V_2], (\sigma_\ell)_{|V_2}) = (\vec{G}_2 \odot F, \sigma_2)$ and $(\vec{G}_2 \odot F, \sigma_2)$ is hourglass-free implies that $(\vec{G}_\ell \odot F, \sigma_\ell)$ must also be hourglass-free. By Prop. 4.8, the BMG $(\vec{G}_\ell \odot F, \sigma_\ell)$ is binary-explainable, and hence, ℓ-BMG DBEG/CBEG/EBEG with input $(\vec{G}_\ell, \sigma_\ell, k)$ has a yes-answer.

For the converse, suppose that ℓ-BMG DBEG/CBEG/EBEG with input $(\vec{G}_\ell, \sigma_\ell, k)$ has a solution $F \subseteq V_\ell \times V_\ell \setminus \{(v,v) \mid v \in V_\ell\}$ with $|F| \leq k$ such that $(\vec{G}_\ell \odot F, \sigma_\ell)$ is a binary-explainable BMG. By Prop. 4.8, $(\vec{G}_\ell \odot F, \sigma_\ell)$ is hourglass-free. As before, let $F' \subseteq F$ be the subset of arc modifications (x,y) for which $x, y \in V_2$. By the same arguments as in the proof of Thm. 5.5, we have $|F'| \leq |F| \leq k$ and $(\vec{G}_2 \odot F', \sigma_2) = ((\vec{G}_\ell \odot F)[V_2], (\sigma_\ell)_{|V_2})$ is a BMG. In particular, $(\vec{G}_2 \odot F', \sigma_2)$ is an induced subgraph of $(\vec{G}_\ell \odot F, \sigma_\ell)$, and thus, hourglass-free. Together with Prop. 4.8, the latter arguments imply that 2-BMG DBEG/CBEG/EBEG with input $(\vec{G}_2, \sigma_2, k)$ has a yes-answer. ☐

We note in passing that 2-BMG Deletion and 2-BMG Completion can be shown to be fixed-parameter tractable (with the number k of edits as parameter) provided that the input digraph is sink-free. To see this, observe that sink-free 2-colored digraphs are BMGs if and only if they do not contain induced F1-, F2-, and F3-subgraphs (cf. Thm. 4.7). The FPT result follows directly from the observation that all such subgraphs are of fixed size and only a fixed number of arc deletions (resp., additions) are possible. In the case of 2-BMG Deletion, only those arc deletions are allowed that do not produce sinks in $\vec{G}$. Clearly, digraphs remain sink-free under arc addition. It remains unclear whether 2-BMG Editing is also FPT for sink-free digraphs. One difficulty is that arc deletions may result in a sink-vertex which then needs to be resolved by subsequent arc additions. It also remains an open question for future research whether the BMG modification problems for (not necessarily sink-free) ℓ-colored digraphs are also FPT. We suspect that this is not the case for $\ell \geq 3$, where the characterization also requires consistency of the set of informative triples. Since removal of a triple from $\mathcal{R}(\vec{G}, \sigma)$ requires the insertion or deletion of an arc, it seems difficult to narrow down the editing candidates to a constant-size set. Indeed, Maximum Triple Inconsistency is not FPT when parametrized by the number k of triples to be excluded [41]. On the other hand, the special case of Dense Maximum Triple Inconsistency is FPT [116]. The set of informative triples $\mathcal{R}(\vec{G}, \sigma)$, however, is usually far from being dense.

In summary, we have shown that arc modification problems for BMGs and beBMG are NP-complete. This is not necessarily an obstacle for using BMG editing in practical workflows – after all, the computational problems in phylogenetics all involve several NP-complete steps, including Multiple Sequence Alignment [86] and the Maximum Parsimony Tree [113] or Maximum Likelihood Tree problems [53]. Nevertheless, highly efficient and accurate heuristics have been devised for these problems, often adjusted to the peculiarities of real-life data, so that the computational phylogenetics have become a routine task in bioinformatics. As a starting point to tackling BMG editing in practice, we will introduce an ILP formulation in the next section that should be workable at least for moderate-size instances, before turning to an in-depth investigation of some heuristic algorithms in Sec. 5.5, aiming to pave the way for larger-scale practical applications.

5.4 ILP FORMULATION OF BMG MODIFICATION PROBLEMS

Hard graph editing problems can often be solved with integer linear programming (ILP) on practically relevant instances. It is of interest, therefore, to consider an ILP formulation of the BMG deletion, completion and editing problems and their binary-explainable BMG counterparts considered above. This section is organized as follows: First, we will describe the encoding of colored digraphs and the objective function together with some constraints that are relevant for all of the considered general and more special cases of BMG modification problems. Then we will turn to additional variables and constraints needed to solve 2-BMG modification problems (based on Thm. 4.7), the general ℓ-BMG modification problems (based on Thm. 4.3), and finally ℓ-BMG DBEG/CBEG/EBEG (based on Thm. 4.4).

As input, we are given an ℓ-colored digraph $(\vec{G} = (V, E), \sigma)$. We encode its arcs by the binary constants

$$E_{xy} = 1 \text{ if and only if } (x, y) \in E \tag{10}$$

for all pairs $(x, y) \in V \times V$, $x \neq y$. The vertex coloring σ is represented by the binary constant

$$\varsigma_{y,s} = 1 \text{ if and only if } \sigma(y) = s \tag{11}$$

for all pairs $(x, s) \in V \times \sigma(V)$. We will denote the modified digraph as $(\vec{G}^*, \sigma)$ and encode its arcs by binary variables ϵ_{xy}, i.e., $\epsilon_{xy} = 1$ if and only if $(x, y) \in E(\vec{G}^*)$. The aim is to minimize the number of edit operations, and thus, the symmetric difference between the respective arc sets. This is represented by the objective function

$$\min \sum_{(x,y)\in V\times V} (1 - \epsilon_{xy})E_{xy} + \sum_{(x,y)\in V\times V} (1 - E_{xy})\epsilon_{xy} \tag{12}$$

with the first sum counting arc deletions and the second counting insertions. The same objective function can also be used for the BMG completion and BMG deletion problem. To ensure that only arcs between vertices of distinct colors exist, we add the constraints

$$\epsilon_{xy} = 0 \text{ for all } (x, y) \in V \times V \text{ with } \sigma(x) = \sigma(y). \tag{13}$$

For the BMG completion problem, the arc set E must be contained in the modified arc set. Hence, we add

$$E_{xy} \leq \epsilon_{xy} \text{ for all } (x, y) \in V \times V. \tag{14}$$

In this case, Eq. (14) ensures that $\epsilon_{xy} = 1$ if $E_{xy} = 1$ and thus, (x, y) remains an arc in the modified digraph. In contrast, for the BMG deletion problem, it is not allowed to add arcs and thus, we use

$$\epsilon_{xy} \leq E_{xy} \text{ for all } (x, y) \in V \times V. \tag{15}$$

In this case, Eq. (15) ensures that $\epsilon_{xy} = 0$ if $E_{xy} = 0$ and thus, (x, y) does not become an arc in the modified digraph. For the BMG editing problem, we neither need Constraint (14) nor (15). The characterizations in Thm. 4.7, 4.3 and Thm. 4.4 require that $(\vec{G}^*, \sigma)$ is sf-colored. Eq. 13 already ensures a proper coloring and thus, it remains to make sure that each vertex has at least one out-neighbor of every other color. This property translates to the constraint

$$\sum_{y \neq x} \epsilon_{xy} \cdot \varsigma_{y,s} > 0 \tag{16}$$

for all $s \neq \sigma(x)$.

The $O(|V|^2)$ variables and $O(|V|^2)$ constraints introduced above are relevant for ℓ-BMG modification problems for an arbitrary ℓ as well as for the modification to a binary-explainable BMG. In the following two subsections, we present additional constraints and variables that are sufficient for the cases $\ell = 2$ and $\ell \geq 2$, respectively.

5.4.1 2-*BMG Modification Problems*

By Thm. 4.7, a properly 2-colored digraph is a BMG if and only if it is sink-free and does not contain an induced F1-, F2-, or F3-graph. Eq. (16) already guarantees that $(\vec{G}^*, \sigma)$ is sink-free. Hence it suffices to add constraints that exclude induced F1-, F2-, and F3-graphs. For every ordered four-tuple $(x_1, x_2, y_1, y_2) \in V^4$ with pairwise distinct x_1, x_2, y_1, y_2 and $\sigma(x_1) = \sigma(x_2) \neq \sigma(y_1) = \sigma(y_2)$, we require

$$\text{(F1)} \qquad \epsilon_{x_1y_1} + \epsilon_{y_1x_2} + \epsilon_{y_2x_2} + (1 - \epsilon_{x_1y_2}) + (1 - \epsilon_{y_2x_1}) \leq 4 \text{ and} \tag{17}$$

$$\text{(F2)} \qquad \epsilon_{x_1y_1} + \epsilon_{y_1x_2} + \epsilon_{x_2y_2} + (1 - \epsilon_{x_1y_2}) \leq 3. \tag{18}$$

In addition, for every ordered five-tuple $(x_1, x_2, y_1, y_2, y_3) \in V^5$ with pairwise distinct x_1, x_2, y_1, y_2, y_3 and $\sigma(x_1) = \sigma(x_2) \neq \sigma(y_1) = \sigma(y_2) = \sigma(y_3)$, we enforce

$$\text{(F3)} \qquad \epsilon_{x_1y_1} + \epsilon_{x_1y_3} + \epsilon_{x_2y_2} + \epsilon_{x_2y_3} + (1 - \epsilon_{x_1y_2}) + (1 - \epsilon_{x_2y_1}) \leq 5. \tag{19}$$

By construction, we still have $O(|V|^2)$ variables but $O(|V|^5)$ constraints. We note that the 2-colored case is handled correctly by the ILP formulation for the general ℓ-colored case given in the next section. However, the additional variables required for $\ell > 2$ are not needed here. We note in passing that, accordingly, we observed a significant speedup when compared to the application of the general formulation to 2-colored digraphs in a cursory simulation.

For the general ℓ-colored case, we drop Equations (17)–(19), and instead rely on Thm. 4.3, which requires that the pair $(\mathcal{R}(\vec{G}^*,\sigma),\mathcal{F}(\vec{G}^*,\sigma))$ is consistent. To implement this constraint, we follow the approach of [47] and [133]. Note that we make no distinction between the two triples $ba|c$ and $ab|c$. In order to avoid superfluous variables and symmetry conditions connecting them, we assume that the first two indices in triple variables are ordered. Thus there are three triple variables $t_{ab|c}$, $t_{ac|b}$ and $t_{bc|a}$ for any three distinct $a,b,c \in V$. We add constraints such that $t_{ab|c} = 1$ if $ab|c$ is an informative triple (cf. Def. 4.4 and Lemma 4.1) and $t_{ab|c} = 0$ if $ab|c$ is a forbidden triple (cf. Def. 4.8 and Lemma 4.12). Hence, we add

$$\epsilon_{xy} + (1-\epsilon_{xy'}) - t_{xy|y'} \leq 1 \text{ and} \tag{20}$$

$$\epsilon_{xy} + \epsilon_{xy'} + t_{xy|y'} \leq 2 \tag{21}$$

for all ordered $(x,y,y') \in V^3$ with three pairwise distinct vertices x,y,y' and $\sigma(x) \neq \sigma(y) = \sigma(y')$. Eq. (20) ensures that if (x,y) is an arc ($\epsilon_{xy} = 1$) and (x,y') is not an arc ($\epsilon_{xy'} = 0$) in the edited digraph, then $t_{xy|y'} = 1$. To obtain a BMG, we must ensure that there is a tree that displays all triples in $\mathcal{R}(\vec{G}^*,\sigma)$ and none of the triples in $\mathcal{F}(\vec{G}^*,\sigma)$.

A phylogenetic tree T is uniquely determined by its sets of clusters $\mathcal{C}(T) := \{L(T(v)) \mid v \in V(T)\}$ [271]. Thus, it is possible to reconstruct T by building the clusters induced by the informative triples while avoiding that forbidden triples are displayed. The set of clusters $\mathcal{C}(T)$ forms a hierarchy, that is, for all $p,q \in \mathcal{C}(T)$ it holds that $p \cap q \in \{\emptyset, p, q\}$. It is easy to see that, in order to recover T from $\mathcal{C}(T)$, it suffices to take into account only the non-trivial clusters $p \in \mathcal{C}(T)$ with $|p| \neq 1$ and $p \neq V(T)$. The number of non-trivial clusters of T is bounded by $L(T) - 2$ (cf. [133, Lemma 1]), where $L(T) = V$, i.e., the vertex set of the ℓ-colored digraph $(\vec{G} = (V,E),\sigma)$ given as input. In order to translate the condition that $\mathcal{C}(T)$ forms hierarchy into the language of ILPs, we follow [47, 133]. Let M be a binary $|V| \times (|V|-2)$ matrix with entries $M(x,p) = 1$ iff vertex $x \in V$ is contained in cluster p. Each cluster p of the tree T_M encoded by M, which is represented by the p-th column of M, corresponds to an inner vertex v_p in T so that $L(T(v_p)) = \{x \mid x \in V, M(x,p) = 1\}$. In the following, we identify column p with the corresponding cluster $L(T(v_p))$.

We next ensure that all informative triples and none of the forbidden triples $ab|c$ are displayed by T_M. This is case if and only if there exists an inner vertex v_p such that $a,b \in L(T(v_p))$ and $c \notin L(T(v_p))$ for every informative triple and no such vertex exists for any forbidden triple. Therefore, we define, for all ordered three-tuples $(a,b,c) \in V^3$ and all $p \in \{1,\dots,|V|-2\}$, the binary variable $m((ab|c),p)$ and set $m((ab|c),p) = 1$ iff $M(a,p) = M(b,p) = 1$ and $M(c,p) = 0$, i.e., iff the cluster p contains a and b but not c. The latter can be achieved by adding, for all these variables, the constraint

$$0 \leq -3 \cdot m((ab|c),p) + M(a,p) + M(b,p) + (1 - M(c,p)) \leq 2. \tag{22}$$

Full enumeration of all possible values that can be assigned to $M(a,p)$, $M(b,p)$ and $M(c,p)$ shows that $m((ab|c),p) = 1$ if and only if $M(a,p) = M(b,p) = 1$ and $M(c,p) = 0$.

For every informative triple $ab|c$ there must be at least one column p for which $m((ab|c),p) = 1$ and for each forbidden triple it must be ensured that $m((ab|c),p) = 0$ for all $p \in \{1,\dots,|V|-2\}$. This is achieved by adding

$$t_{ab|c} \leq \sum_{p=1}^{|V|-2} m((ab|c),p) \leq (|V|-2) \cdot t_{ab|c} \tag{23}$$

for all ordered $(a,b,c) \in V^3$ with three pairwise distinct vertices a,b,c and $\sigma(a) \neq \sigma(b) = \sigma(c)$. If $t_{ab|c} = 1$, then $m((ab|c),p) = 1$ for at least one p and if $t_{ab|c} = 0$ then, $\sum_{p=1}^{|V|-2} m((ab|c),p) \leq 0$ implies that all $m((ab|c),p)$ are put to 0.

Finally, we must ensure that the matrix M indeed encodes the hierarchy of a tree. This is the case if all clusters p and q are compatible, i.e., if $p \cap q \in \{p,q,\emptyset\}$. Equivalently, two clusters p and q are incompatible if there are vertices a, b and c such that $a \in p \setminus q$, $b \in q \setminus p$ and $c \in p \cap q$, which is represented by the "gametes" $(M(a,p), M(a,q)) = (1,0)$, $(M(b,p), M(b,q)) = (0,1)$ and $(M(c,p), M(c,q)) = (1,1)$. We avoid such incompatible clusters by using the so-called three-gamete condition which is described e.g. in [119] or [133, SI]. To this end, we add for each of the three gametes $(\Gamma,\Lambda) \in \{(0,1),(1,0),(1,1)\}$ the binary variables $C(p,q,\Gamma\Lambda)$ for every pair of columns $p \neq q$. Furthermore, we add the constraints

$$C(p,q,01) \geq -M(a,p) + M(a,q) \tag{24}$$

$$C(p,q,10) \geq M(a,p) - M(a,q) \tag{25}$$

$$C(p,q,11) \geq M(a,p) + M(a,q) - 1 \tag{26}$$

for every pair of columns $p \neq q$ and every $a \in V$. This ensures that $C(p,q,\Gamma\Lambda) = 1$ whenever $M(a,p) = \Gamma$ and $M(a,q) = \Lambda$ holds for at least one $a \in V$. Finally, we add the constraint

$$C_{p,q,01} + C_{p,q,10} + C_{p,q,11} \leq 2 \tag{27}$$

for every pair of columns $p \neq q$, in order to ensure the compatibility of clusters p and q.

In total, this ILP formulation requires $O(|V|^4)$ variables and $O(|V|^4)$ constraints where the most expensive part stems from the variables $m((ab|c),p)$ and their corresponding constraints (cf. Eq. 22).

5.4.3 *Binary-Explainable BMG Modification Problems*

For solving the problems ℓ-BMG DBEG, ℓ-BMG CBEG and ℓ-BMG EBEG with an arbitrary number of colors $\ell \geq 2$, we also require Eqs. (10)–(16) (depending on the individual of the three problems as described above). Additionally, we can employ Thm. 4.4, which states that the properly vertex-colored digraph $(\vec{G}^*, \sigma)$ is a binary-explainable BMG if and only if (i) $(\vec{G}, \sigma)$ is sf-colored, and (ii) $\mathcal{R}^{\mathrm{B}}(\vec{G}^*, \sigma)$ is consistent. Condition (i) is already satisfied by Eqs. (13) and (16). To implement Condition (ii), i.e., consistency of $\mathcal{R}^{\mathrm{B}}(\vec{G}^*, \sigma)$, we follow the approach of [133]. As before, no distinction is made between two triples $ba|c$ and $ab|c$, and we assume that the first two indices in triple variables are ordered, i.e., there are three triple variables $t_{ab|c}$, $t_{ac|b}$ and $t_{bc|a}$ for any three distinct $a, b, c \in V$. We add constraints such that $t_{ab|c} = 1$ if $ab|c \in \mathcal{R}^{\mathrm{B}}(\vec{G}^*, \sigma)$ (cf. Eq. (7)), i.e.,

$$\epsilon_{xy} + (1 - \epsilon_{xy'}) - t_{xy|y'} \leq 1 \tag{28}$$

$$\epsilon_{xy} + \epsilon_{xy'} - t_{yy'|x} \leq 1 \tag{29}$$

for all ordered $(x, y, y') \in V^3$ with three pairwise distinct vertices x, y, y' and $\sigma(x) \neq \sigma(y) = \sigma(y')$. Eq. (28) ensures that if $xy|y'$ is an informative triple, i.e., (x, y) is an arc ($\epsilon_{xy} = 1$) and (x, y') is not an arc ($\epsilon_{xy'} = 0$) in the edited digraph, then $t_{xy|y'} = 1$. Similarly, Eq. (29) ensures that if $xy|y'$ and $xy'|y$ are forbidden triples, i.e., $\epsilon_{xy} = 1$ and $\epsilon_{xy'} = 1$, then $t_{yy'|x} = 1$. These constraints allow some degree of freedom for the choice of the binary triple variables. For example, we may put $t_{xy|y'} = 1$ also in case (x, y) is not an arc. However, by Lemma 2.1, for every consistent set of triples $\mathcal{R}$ on V, there is a strictly dense consistent set of triples $\mathcal{R}'$ with $\mathcal{R} \subseteq \mathcal{R}'$. We therefore add the constraint

$$t_{ab|c} + t_{ac|b} + t_{bc|a} = 1 \text{ for all } \{a, b, c\} \in \binom{V}{3} \tag{30}$$

that ensures that precisely one of the binary variables representing one of the three possible triples on three leaves is set to 1. The final set $\mathcal{R}'$ of triples obtained in this manner contains all informative triples but could be larger than $\mathcal{R}^{\mathrm{B}}(\vec{G}, \sigma)$. This reflects Thm. 4.5 which states that every binary tree (T', σ') explaining a BMG $(\vec{G}', \sigma')$ is a refinement of the BRT (T, σ'). Moreover, note that the triple set $r(T')$ of the binary tree T' is clearly strictly dense. In particular, we have $\mathcal{R}^{\mathrm{B}}(\vec{G}', \sigma') \subseteq r(T) \subseteq r(T')$.

To ensure consistency of the triple set, we employ Thm. 1, Lemma 4, and ILP 5 from [133], which are based on so-called 2-order inference rules and add

$$2t_{ab|c} + 2t_{ad|b} - t_{bd|c} - t_{ad|c} \leq 2 \text{ for all } \{a, b, c, d\} \in \binom{V}{4}. \tag{31}$$

In summary, we require $O(|V|^3)$ variables and $O(|V|^4)$ conditions, where the most expensive parts are the triple variables $t_{ab|c}$ and the 2-order inference rules in Eq. (31). For comparison, the general approach above requires $O(|V|^4)$ variables and conditions. Due to the lower number of possible choices for the variables, we expect the ILP solution for the binary-explainable-restricted case to run (at least moderately) faster.

5.4.4 *Remarks on the Running Time*

We tested all three versions of BMG editing, i.e., the 2-colored, the general ℓ-colored and the binary-explainable-restricted ℓ-colored approach, using IBM

ILOG CPLEX™ Optimizer 12.10 [154] and Gurobi Optimizer 9.0 [118], and applied them to randomly disturbed 2-BMGs, BMGs in general and binary-explainable BMGs. Without further optimization, instances with $\ell \geq 3$ colors are very demanding and already problems with ten vertices may take a few hours on a desktop system. The 2-colored version (requiring only $O(|V|^2)$ variables) on the other hand, handles instances with 20 vertices in about a minute. Not surprisingly, the version for beBMG editing also outperforms the general ℓ-colored version in terms of speed. However, it was only able to process instances of up to 15 vertices in about a minute. We note that if one aims to obtain a beBMG from a 2-colored digraph, one could amend the general 2-colored version in Sec. 5.4.1 by additional $O(|V|^4)$ constraints that forbid hourglasses in the editing result $(\vec{G}^*, \sigma)$, which may lead to a slight speed-up for this special case.

5.5 HEURISTICS OF BEST MATCH GRAPH EDITING

In the previous sections, we have seen that the arc modification problems for BMGs are all NP-complete in general, and provided ILP solutions allowing exact solutions for small instances. However, in computational biology, applications to large gene families would be of particular interest, creating the need for faster, approximate solutions for BMG editing. Before embarking to develop software for a BMG-based analysis of large sequence data sets, we need to understand whether the editing problem for BMGs is tractable in practice with sufficient accuracy and for interestingly large instances.

Motivated by both theoretical and practical considerations, we are mainly interested in heuristics that are *consistent* in the sense that they leave input digraphs $(\vec{G}, \sigma)$ that are already BMGs unchanged. More precisely, the heuristic algorithms considered in this section can be thought of as maps $\mathbb{A}$ on the set of finite vertex-colored digraphs such that $\mathbb{A}(\vec{G}, \sigma)$ is a BMG for every vertex-colored input digraph $(\vec{G}, \sigma)$. In particular, the following property of such algorithms is desirable:

Definition 5.4. *A (BMG-editing) algorithm is* consistent *if* $\mathbb{A}(\vec{G}, \sigma) = (\vec{G}, \sigma)$ *whenever* $(\vec{G}, \sigma)$ *is a BMG.*

Similarly, we will call a heuristic for the MAXIMUM ROOTED TRIPLE CONSISTENCY PROBLEM (MAXRTC), see Sec. 5.5.1 below, consistent if it returns the full set $\mathcal{R}$ whenever the input $\mathcal{R}$ is already a consistent set of triples.

By Thm. 4.2, the LRT of a BMG $(\vec{G} = (V, E, \sigma)$ can be constructed from its set of informative triples as $\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma), V)$ by means of the `BUILD` algorithm. The simplest approach, therefore, is to extract a maximal consistent subset $\mathcal{R}^*$ from $\mathcal{R}(\vec{G}, \sigma)$ and to use $\vec{G}(\mathrm{Aho}(\mathcal{R}^*, V), \sigma)$ as an approximation, see Sec. 5.5.1.

A more detailed analysis of arcs in $(\vec{G}, \sigma)$ that violate the property of being a BMG in Sec. 5.5.2, however, will lead to a notion of "unsatisfiable relations" (UR), which can be used to count the arc modifications associated with a partition $\mathcal{V}$ of the vertex set V of $\vec{G}$. It also gives rise to a top-down algorithm in which the vertex set of $\vec{G}$ is recursively edited and partitioned. A large class of heuristics for BMG editing can be constructed depending on the construction of the partition $\mathcal{V}$ in each recursion step. We shall see that the arc edit sets in different steps of the recursion are disjoint. A main result of this section, Thm. 5.8, links the partitions $\mathcal{V}$ appearing in BMG editing algorithms to the auxiliary graphs appearing in the `BUILD` algorithm for supertree construction [7]. This provides a guarantee that the BMG editing algorithms are consistent provided the choice of $\mathcal{V}$ is such that it does not enforce edits whenever an alternative partition with empty UR is available. For BMGs, this is in particular the case for the partitions appearing in the `BUILD` algorithm. In Sec. 5.5.3, we proceed to show by reduction from Set Splitting that finding a partition with a minimal number of unsatisfiable relations is NP-hard.

The theoretical results are complemented by computational experiments on BMGs with randomly perturbed arc sets in Sec. 5.5.4. We focus on a comparison of different algorithms to construct the partitions $\mathcal{V}$. Somewhat surprisingly, we find that minimizing the cardinality of the UR alone is not the best approach, since this tends to produce very unbalanced partitions and thus requires a large number of steps in the recursions whose costs add up. Instead, certain types of clustering or community detection approaches that favor more balanced partitions tend to perform well.

5.5.1 *A Simple, Triple-Based Heuristic*

By Thm. 4.2, a vertex-colored digraph $(\vec{G} = (V, E), \sigma)$ is a BMG if and only if (a) its set of informative triples $\mathcal{R}(\vec{G}, \sigma)$ is consistent and (b) the BMG $\vec{G}(\hat{T}, \sigma)$ of the corresponding tree $\hat{T} := \text{Aho}(\mathcal{R}(\vec{G}, \sigma), V)$ coincides with $(\vec{G}, \sigma)$. In general, the tree $\text{Aho}(\mathcal{R}, V)$ of a consistent set of triples $\mathcal{R}$ on a set V is a least resolved supertree of all the triples in $\mathcal{R}$. For a BMG, $\hat{T} := \text{Aho}(\mathcal{R}(\vec{G}, \sigma), V)$ is the unique least resolved tree (LRT) for $(\vec{G}, \sigma) = \vec{G}(\hat{T}, \sigma)$. These close connections between recognizing BMGs and constructing supertrees suggest to adapt ideas from heuristic algorithms for triple consistency problems and supertree construction for BMG editing. In particular, the triple set $\mathcal{R}(\vec{G}, \sigma)$ can easily be read off an arbitrary vertex-colored input digraph.

We expect that empirically estimated best match relations will typically contain errors that correspond to both arc insertions and deletions w.r.t. the unknown underlying "true" best match graph. It is important to note that both arc insertions and deletions may lead to creation and loss of informative triples. In particular, when starting from a BMG, both types of modifications have the potential to make the triple set inconsistent as the example in Fig. 30 shows. This is indeed often the case even for moderate disturbances of a BMG as we shall see in Sec. 5.5.4.

The latter argumentation suggests a simple heuristic for BMG editing that relies on replacing the consistency checks for triple sets by the extraction of maximal sets of consistent triples $\mathcal{R}^*$ from $\mathcal{R}(\vec{G}, \sigma)$ and to use the BMG $\vec{G}(\text{Aho}(\mathcal{R}^*, V), \sigma)$ as an approximation, see Alg. 4.

Unfortunately, the decision versions of both MaxRTC, the problem of extracting from a given set $\mathcal{R}$ of rooted triples a maximum-size consistent subset, and MinRTI, the problem of finding a minimum-size subset $\mathcal{I}$ such that $\mathcal{R} \setminus \mathcal{I}$ is consistent, are themselves NP-hard [157]. Furthermore, MaxRTC is APX-

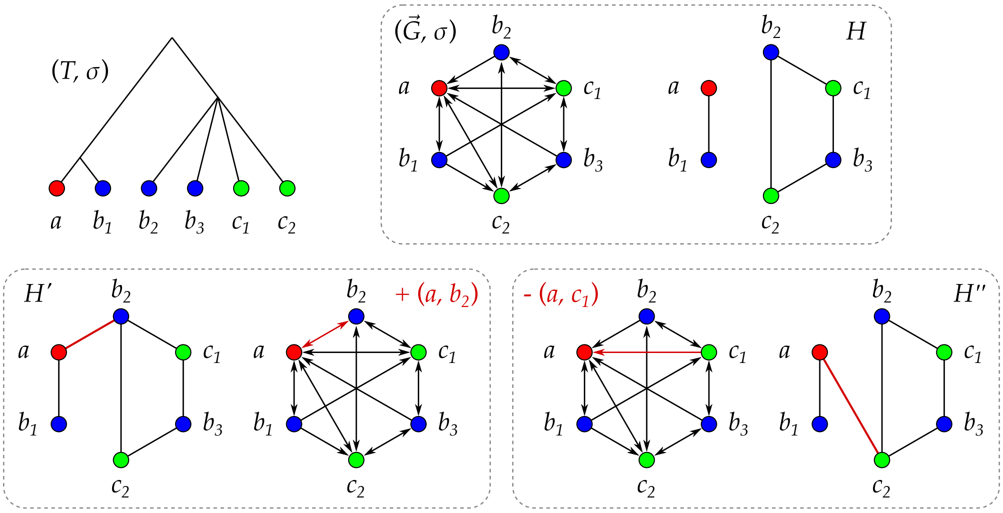


Fig. 30. Both arc insertions and deletions into a BMG $(\vec{G} = (V,E),\sigma)$ can introduce inconsistencies into the set of informative triples. Top row: Leaf-colored tree (T,σ) explaining the BMG $(\vec{G},\sigma)$. Its set of informative triples is $\mathcal{R}(\vec{G},\sigma) = \{ab_1|b_2, ab_1|b_3, c_1b_2|b_1, c_1b_3|b_1, c_2b_2|b_1, c_2b_3|b_1\}$ giving the Aho graph $H = [\mathcal{R}(\vec{G},\sigma), V]$. Bottom left: Insertion of the arc (a,b_2) creates a new informative triple $ab_2|b_3$ ($ab_1|b_2$ gets lost) resulting in a connected Aho graph H'. Bottom right: Deletion of the arc (a,c_1) creates a new triple $ac_2|c_1$ resulting in a connected Aho graph H''.

hard and MinRTI is $\Omega(\ln n)$-inapproximable [41]. However, because of their practical importance in phylogenetics, a large number of practically useful heuristics have been devised, see e.g. [100, 312, 333].

Algorithm 4: Simple Heuristic for BMG editing.

```
Input: Properly colored digraph $(\vec{G},\sigma)$.
Output: BMG $(\vec{G}^*,\sigma)$.
$\mathcal{R}^* \leftarrow$ MaxRTC$(\mathcal{R}(\vec{G},\sigma))$
return $\vec{G}(\mathrm{Aho}(\mathcal{R}^*, V(\vec{G})),\sigma)$
```

As a consequence of Thm. 4.2, Alg. 4 is consistent, i.e., $(\vec{G}^*,\sigma) = (\vec{G},\sigma)$ if and only if the input digraph $(\vec{G},\sigma)$ is a BMG, if a consistent heuristic is employed to solve MaxRTG/MinRTI, i.e., if consistent triple sets remain unchanged by the method approximating MaxRTG/MinRTI.

The heuristic Alg. 4 is not always optimal, even if MaxRTC/MinRTI is solved optimally. Fig. 31 shows an unconnected 2-colored digraph $(\vec{G},\sigma)$ on three vertices that is not a BMG and does not contain informative triples. The BMG $(\vec{G}^*,\sigma)$ produced by Alg. 4 introduces two arcs into $(\vec{G},\sigma)$. However, $(\vec{G},\sigma)$ can also be edited to a BMG by inserting only one arc.

A simple improvement is to start by enforcing obvious arcs: If v is the only vertex with color $\sigma(v)$, then by definition there must be an arc (x,v) for every vertex $x \neq v$. The computation then starts from the sets of informative triples of the modified digraph. We shall see that these are the only arcs that can safely be added to $\vec{G}$ without other additional knowledge or constraints (cf. Thm. 5.7 below).

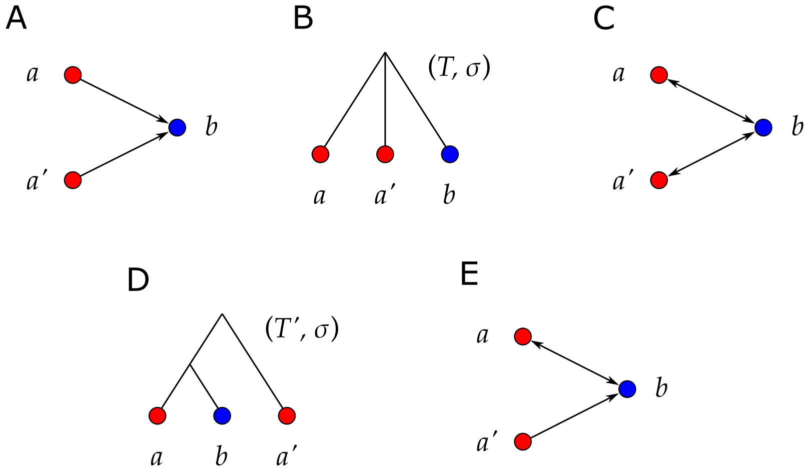


Fig. 31. Example for a digraph (A) where Alg. 4 does not lead to an optimal BMG editing. The set $\mathcal{R}(\vec{G},\sigma)$ is empty and thus consistent. (B) The tree $T = \text{Aho}(\mathcal{R}(\vec{G},\sigma), V(\vec{G}))$ and (C) its corresponding BMG. The two arcs (b,a) and (b,a') have been inserted. (D) A tree (T',σ) and (E) its corresponding BMG $\vec{G}(T',\sigma)$ in which only the arc (b,a) has been inserted.

5.5.2 *Locally Optimal Splits*

Finding an optimal BMG editing of a digraph $(\vec{G} = (V,E),\sigma)$ is equivalent to finding a tree (T,σ) on V that minimizes the cardinality of

$$\begin{aligned} U(\vec{G},T) := \{(x,y) \in V \times V \mid (x,y) \in E \text{ and } (x,y) \notin E(\vec{G}(T,\sigma)), \text{ or} \\ (x,y) \notin E \text{ and } (x,y) \in E(\vec{G}(T,\sigma))\}. \end{aligned} \tag{32}$$

Clearly, $U(\vec{G},T) = \emptyset$ implies that $(\vec{G},\sigma) = \vec{G}(T,\sigma)$ is a BMG. However, finding a tree (T,σ) that minimizes $|U(\vec{G},T)|$ is intractable (unless $P = NP$) since ℓ-BMG Editing, Problem 5.2 above, is NP-complete [266].

We may ask, nevertheless, if trees (T,σ) on V contain information about arcs and non-arcs in $(\vec{G},\sigma)$ that are "unambiguously false" in the sense that they are contained in every edit set that converts $(\vec{G},\sigma)$ into a BMG. Denote by $\mathcal{T}_V$ the set of all phylogenetic trees on V. The set of these "unambiguously false" (non-)arcs can then be expressed as

$$U^*(\vec{G}) := \bigcap_{T\in\mathcal{T}_V} U(\vec{G},T). \tag{33}$$

Since there are in general exponentially many trees on V and thus, the problem of determining $U^*(\vec{G})$ seems to be quite challenging on a first glance. We shall see in Thm. 5.7, however, that $U^*(\vec{G})$ can be computed efficiently. We start with a conceptually simpler construction.

Definition 5.5. *Let $(\vec{G} = (V,E),\sigma)$ be a properly vertex-colored digraph and $\mathcal{V}$ a partition of V with $|\mathcal{V}| \geq 2$. Moreover, let $\mathcal{T}(\mathcal{V})$ be the set of trees T on V that satisfy $\mathcal{V} = \{L(T(v)) \mid v \in \mathrm{child}_T(\rho_T)\}$. The set of* unsatisfiable relations *(UR), denoted by $U(\vec{G},\mathcal{V})$, is defined as*

$$U(\vec{G},\mathcal{V}) := \bigcap_{T\in\mathcal{T}(\mathcal{V})} U(\vec{G},T). \tag{34}$$

The associated UR-cost is $c(\vec{G},\mathcal{V}) := |U(\vec{G},\mathcal{V})|$.

The set of (phylogenetic) trees $\mathcal{T}(\mathcal{V})$ is non-empty since $|\mathcal{V}| \geq 2$ in Def. 5.5. Moreover, by construction, $(x,y) \in U(\vec{G},\mathcal{V})$ if and only if

$$\begin{aligned}&(x,y) \in E \text{ and } (x,y) \notin E(\vec{G}(T,\sigma)) \text{ for all } T \in \mathcal{T}(\mathcal{V})\text{, or}\\ &(x,y) \notin E \text{ and } (x,y) \in E(\vec{G}(T,\sigma)) \text{ for all } T \in \mathcal{T}(\mathcal{V}).\end{aligned}$$

Intriguingly, the set $U(\vec{G},\mathcal{V})$, and thus the *UR*-cost $c(\vec{G},\mathcal{V})$, can be computed in polynomial time without any explicit knowledge of the possible trees to determine the set $U(\vec{G},\mathcal{V})$. To this end, we define the three sets

$$\begin{aligned}U_1(\vec{G},\mathcal{V}) &= \bigcup_{V_i\in\mathcal{V}} \{(x,y) \mid (x,y) \in E,\ x \in V_i,\ y \in V\setminus V_i,\ \sigma(y) \in \sigma(V_i)\},\\ U_2(\vec{G},\mathcal{V}) &= \bigcup_{V_i\in\mathcal{V}} \{(x,y) \mid (x,y) \notin E,\ x \in V_i,\ y \in V\setminus V_i,\ \sigma(y) \notin \sigma(V_i)\},\\ U_3(\vec{G},\mathcal{V}) &= \bigcup_{V_i\in\mathcal{V}} \{(x,y) \mid (x,y) \notin E,\ \text{distinct } x,y \in V_i,\ V_i[\sigma(y)] = \{y\}\}.\end{aligned}$$

Lemma 5.6. *Let* $(\vec{G} = (V,E),\sigma)$ *be a properly vertex-colored digraph and let* $\mathcal{V} = \{V_1,\dots,V_k\}$ *be a partition of* V *with* $|\mathcal{V}| = k \geq 2$. *Then*

$$U(\vec{G},\mathcal{V}) = U_1(\vec{G},\mathcal{V}) \ \charfusion[\mathbin]{\cup}{\cdot}\ U_2(\vec{G},\mathcal{V}) \ \charfusion[\mathbin]{\cup}{\cdot}\ U_3(\vec{G},\mathcal{V})\,.$$

Proof. We first note that $U_1 := U_1(\vec{G},\mathcal{V})$, $U_2 := U_2(\vec{G},\mathcal{V})$ and $U_3 := U_3(\vec{G},\mathcal{V})$ are pairwise disjoint. Furthermore, we have $x \neq y$ and $\sigma(x) \neq \sigma(y)$ for every $(x,y) \in U_1 \,\dot\cup\, U_2 \,\dot\cup\, U_3$ and every $(x,y) \in U(\vec{G},\mathcal{V})$. Moreover, recall that $\mathcal{T}(\mathcal{V})$ is the set of trees T on V that satisfy $\mathcal{V} = \{L(T(v)) \mid v \in \mathsf{child}_T(\rho_T)\}$. Therefore, there is a one-to-one correspondence between the $k \geq 2$ sets in $\mathcal{V}$ and the children $\mathsf{child}_T(\rho_T)$ of the root ρ_T for any $T \in \mathcal{T}(\mathcal{V})$. We denote by v_i the child corresponding to $V_i \in \mathcal{V}$; thus $V_i = L(T(v_i))$.

We first show that $(x,y) \in U_1 \,\dot\cup\, U_2 \,\dot\cup\, U_3$ implies $(x,y) \in U(\vec{G},\mathcal{V})$. Let $T \in \mathcal{T}(\mathcal{V})$ be chosen arbitrarily, and let ρ be its root. Suppose that $(x,y) \in U_1$. Thus, we have $(x,y) \in E$, $\sigma(y) \in \sigma(L(T(v_i)))$, $x \preceq_T v_i$ and $y \preceq_T v'$ for some $v' \in \mathsf{child}_T(\rho) \setminus \{v_i\}$. Moreover, $\sigma(y) \in \sigma(L(T(v_i)))$ implies that there is a vertex $y' \preceq_T v_i$ with $\sigma(y') = \sigma(y)$. Taken together, we obtain $\mathrm{lca}_T(x,y') \preceq_T v_i \prec_T \rho = \mathrm{lca}_T(x,y)$, and thus $(x,y) \notin E(\vec{G}(T,\sigma))$. If $(x,y) \in U_2$, we have $(x,y) \notin E$, $\sigma(y) \notin \sigma(L(T(v_i)))$, $x \preceq_T v_i$ and $y \preceq_T v'$ for some $v' \in \mathsf{child}_T(\rho) \setminus \{v_i\}$. Moreover, $\sigma(y) \notin \sigma(L(T(v_i)))$ implies that there is no vertex $y' \preceq_T v_i$ with $\sigma(y') = \sigma(y)$. Thus, $\mathrm{lca}_T(x,y') = \mathrm{lca}_T(x,y) = \rho$ holds for all y' of color $\sigma(y') = \sigma(y)$, and thus $(x,y) \in E(\vec{G}(T,\sigma))$. Finally, suppose $(x,y) \in U_3$. We have $(x,y) \notin E$ and y is the only leaf of its color in $L(T(v_i))$. Therefore, there is no vertex y' with $\sigma(y') = \sigma(y)$ and $\mathrm{lca}_T(x,y') \prec_T \mathrm{lca}_T(x,y)$, and thus $(x,y) \in E(\vec{G}(T,\sigma))$. In summary, one of the conditions in Def. 5.5 is satisfied for T in all three cases. Since T was chosen arbitrarily, we conclude $(x,y) \in U(\vec{G},\mathcal{V})$ for any $(x,y) \in U_1 \,\dot\cup\, U_2 \,\dot\cup\, U_3$.

In order to show that $(x,y) \in U(\vec{G},\mathcal{V})$ implies $(x,y) \in U_1 \,\dot\cup\, U_2 \,\dot\cup\, U_3$, we distinguish *Case (a)*: $(x,y) \in E$ and $(x,y) \notin E(\vec{G}(T,\sigma))$ holds for all $T \in \mathcal{T}(\mathcal{V})$, and *Case (b)*: $(x,y) \notin E$ and $(x,y) \in E(\vec{G}(T,\sigma))$ holds for all $T \in \mathcal{T}(\mathcal{V})$.

Case (a). $(x,y) \in E$ implies $\sigma(x) \neq \sigma(y)$. Moreover, there is a vertex y' with $\sigma(y') = \sigma(y)$, and $\mathrm{lca}_T(x,y') \prec_T \mathrm{lca}_T(x,y)$ for every $T \in \mathcal{T}(\mathcal{V})$ because $(x,y) \notin E(\vec{G}(T,\sigma))$. Since this is true for all trees in $\mathcal{T}(\mathcal{V})$, there must be a set $V_i \in \mathcal{V}$ such that $x,y' \in V_i$, and in particular $\sigma(y') = \sigma(y) \in \sigma(V_i)$. Now suppose, for contradiction, that

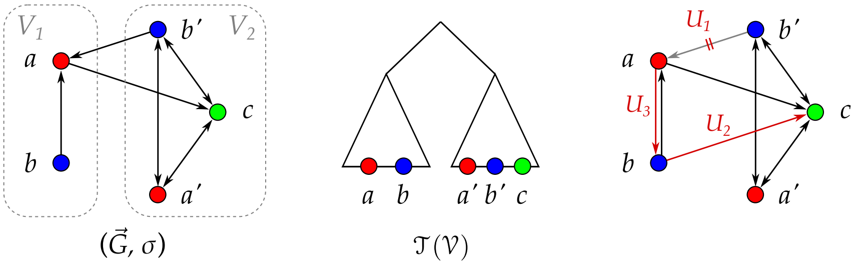


Fig. 32. Example for unsatisfiable relations $U(\vec{G},\mathcal{V})$ of a vertex-colored digraph $(\vec{G}=(V,E),\sigma)$ w.r.t. a partition $\mathcal{V}=\{V_1,V_2\}$ (indicated by the gray boxes). In the middle, the set of trees $\mathcal{T}(\mathcal{V})$ is illustrated, i.e., the triangles represent all possible phylogenetic trees on the respective subset of leaves. On the right, the arc modifications implied by $\mathcal{V}$ (i.e., $U(\vec{G},\mathcal{V})$) are illustrated where U_1, U_2, and U_3 indicate the type according to Lemma 5.6.

$y \in V_i$ and thus $x,y,y' \in V_i$. In this case, we can choose a tree $T \in \mathcal{T}(\mathcal{V})$ such that $x,y \prec_T v_i$ for some child $v_i \in \mathsf{child}_T(\rho_T)$ and $\mathsf{lca}_T(x,y) \preceq_T \mathsf{lca}_T(x,y')$ hold for all y' of color $\sigma(y')=\sigma(y)$. Hence, we obtain $(x,y)\in E(\vec{G}(T,\sigma))$ for this tree; a contradiction. Therefore, we conclude that $y \in V\setminus V_i$. In summary, all conditions for U_1 are satisfied, and thus $(x,y)\in U_1$.
Case (b). We have $(x,y)\notin E$ and $(x,y)\in E(\vec{G}(T,\sigma))$ for all $T\in\mathcal{T}(\mathcal{V})$. Let $V_i\in\mathcal{V}$ such that $x\in V_i$. We distinguish the two cases (i) $y\notin V_i$, and (ii) $y\in V_i$. In Case (i), suppose, for contradiction, that $\sigma(y)\in\sigma(V_i)$. Then, for every tree $T\in\mathcal{T}(\mathcal{V})$, there must be a vertex y' of color $\sigma(y)$ such that $\mathsf{lca}_T(x,y')\preceq_T v_i \prec_T \rho_T = \mathsf{lca}_T(x,y)$, contradicting $(x,y)\in E(\vec{G}(T,\sigma))$. Therefore, we conclude $\sigma(y)\notin\sigma(V_i)$. It follows that $(x,y)\in U_2$. In Case (ii), assume, for contradiction, that there is a vertex $y'\in V_i$ of color $\sigma(y)$ such that $y\neq y'$. This together with $\sigma(y')=\sigma(y)\neq\sigma(x)$ implies that all three vertices x,y,y' are pairwise distinct. Since in addition $x,y,y'\in V_i$, we can choose a tree $T\in\mathcal{T}(\mathcal{V})$ such that $x,y,y'\prec_T v_i$ for some child $v_i\in\mathsf{child}_T(\rho_T)$ and $\mathsf{lca}_T(x,y')\prec_T \mathsf{lca}_T(x,y)$; a contradiction to $(x,y)\in E(\vec{G}(T,\sigma))$ for all $T\in\mathcal{T}(\mathcal{V})$. Therefore, we conclude that y is the only vertex of its color in V_i. It follows that $(x,y)\in U_3$. ☐

The proof of Lemma 5.6 relates the possible cases between $\mathcal{V}$ and the tree set $\mathcal{T}(\mathcal{V})$ in a straightforward manner. Fig. 32 gives examples for all three types of unsatisfiable relations, i.e., for $U_1(\vec{G},\mathcal{V})$, $U_2(\vec{G},\mathcal{V})$, and $U_3(\vec{G},\mathcal{V})$. In particular, we have $(b',a)\in U_1(\vec{G},\mathcal{V})$ since it is an arc in $\vec{G}$ but V_2 contains another red vertex a'. Moreover, $(b,c)\in U_2(\vec{G},\mathcal{V})$ since it is not an arc in $\vec{G}$ but V_1 does not contain another green vertex. Finally, we have $(a,b)\in U_3(\vec{G},\mathcal{V})$ since it is not an arc in $\vec{G}$ but b is the only blue vertex in V_1. In the example, the digraph $(\vec{G}\,\triangle\, U(\vec{G},\mathcal{V}))$ is already a BMG which, however, is not true in general.

Corollary 5.7. *The set $U(\vec{G},\mathcal{V})$ can be computed in quadratic time.*

Proof. We first compute all numbers $n_{i,A}$ of vertices in V_i with a given color A. This can be done in $O(|V|)$ if we do not explicitly store the zero-entries. Now, $\sigma(y)\in\sigma(V_i)$, i.e. $n_{i,\sigma(y)}>0$, can be checked in constant time, and thus, it can also be decided in constant time whether or not a pair (x,y) is contained in $U_1(\vec{G},\mathcal{V})$ or $U_2(\vec{G},\mathcal{V})$. Since, given $y\in V_i$, the condition $V_i[\sigma(y)]=\{y\}$ is equivalent to $n_{i,\sigma(y)}=1$, membership in $U_3(\vec{G},\mathcal{V})$ can also be decided in constant time. Checking all ordered pairs $x,y\in V$ thus requires a total effort of $O(|V|^2)$. ☐

Algorithm 5: General BMG editing via locally optimal steps.
The parts highlighted in color produce a tree T explaining the edited digraph $(\vec{G}^*, \sigma)$. If the tree is not needed, these steps can be omitted. The method for choosing the partition $\mathcal{V}$ (framed box) determines different variants of the algorithm.

```
   Input: Properly colored digraph $(\vec{G} = (V, E), \sigma)$.
   Output: BMG $(\vec{G}^*, \sigma)$.
1  initialize $(\vec{G}^*, \sigma) \leftarrow (\vec{G}, \sigma)$
2  Function Edit($V'$)
3      if $|V'| > 1$ then
4          $\mathcal{V} \leftarrow \boxed{\text{suitably chosen partition of } V' \text{ with } |\mathcal{V}| \geq 2}$
5          $\vec{G}^* \leftarrow \vec{G}^* \triangle U(\vec{G}^*[V'], \mathcal{V})$
6          create a tree $T'$ with root $\rho'$
7          foreach $V_i \in \mathcal{V}$ do
8              attach the tree Edit($V_i$) to $\rho'$
9          return $T'$
10     else
11         return a tree with the single element in $V'$ as root
12 $T \leftarrow$ Edit($V(\vec{G})$)
13 return $(\vec{G}^*, \sigma)$ and $T$
```

Our discussion so far suggests a recursive top-down approach, made precise in Alg. 5. In each step, one determines a “suitably chosen” partition $\mathcal{V}$ and then recurses on the subgraphs of the edited digraph $\vec{G}^* \triangle U(\vec{G}^*[V'], \mathcal{V})$. More details on such suitable partitions $\mathcal{V}$ will be given in Thm. 5.8 below. The parts in the algorithm highlighted in color can be omitted. They are useful, however, if one is also interested in a tree (T, σ) that explains the editing result $(\vec{G}^*, \sigma)$ and to show that $(\vec{G}^*, \sigma)$ is indeed a BMG (see below).

Alg. 5 is designed to accumulate the edit sets in each step, Line 5. In particular, the total edit cost and the scores $c(\vec{G}^*[V'], \mathcal{V})$ are closely tied together, which follows from Lemma 5.8 below. For its proof, we first need the following technical result which shows that the editing of an arc in Alg. 5 will not be reversed in the subsequent recursion step.

Lemma 5.7. *Let $(\vec{G} = (V, E), \sigma)$ be a properly vertex-colored digraph, $\mathcal{V} = \{V_1, \dots, V_k\}$ a partition of V with $|\mathcal{V}| = k \geq 2$, and $\mathcal{V}_i = \{V_{i,1}, \dots, V_{i,l}\}$, $1 \leq i \leq k$, a partition of V_i with $|\mathcal{V}_i| = l \geq 2$. Moreover, let $(\vec{G}' := \vec{G} \triangle U(\vec{G}, \mathcal{V}), \sigma)$ be the colored digraph that is obtained by applying the edits in $U(\vec{G}, \mathcal{V})$ to $(\vec{G}, \sigma)$. Then $U(\vec{G}, \mathcal{V}) \cap U(\vec{G}'[V_i], \mathcal{V}_i) = \emptyset$.*

Proof. Let $\vec{G}'_i := \vec{G}'[V_i]$. The sets of unsatisfiable relations $U(\vec{G}, \mathcal{V})$ and $U(\vec{G}'_i, \mathcal{V}_i)$ are given by the (disjoint) unions $U_1(\vec{G}, \mathcal{V}) \mathbin{\dot\cup} U_2(\vec{G}, \mathcal{V}) \mathbin{\dot\cup} U_3(\vec{G}, \mathcal{V})$ and $U_1(\vec{G}'_i, \mathcal{V}_i) \mathbin{\dot\cup} U_2(\vec{G}'_i, \mathcal{V}_i) \mathbin{\dot\cup} U_3(\vec{G}'_i, \mathcal{V}_i)$, respectively (cf. Lemma 5.6). First, let $(x, y) \in U_1(\vec{G}, \mathcal{V})$. Since, by definition of $U_1(\vec{G}, \mathcal{V})$, x and y are contained in different sets of the partition $\mathcal{V}$, they cannot be both contained in V_i and thus, $U_1(\vec{G}, \mathcal{V}) \cap U(\vec{G}'_i, \mathcal{V}_i) = \emptyset$. One analogously argues that $U_2(\vec{G}, \mathcal{V}) \cap U(\vec{G}'_i, \mathcal{V}_i) = \emptyset$. Now, assume for contradiction that $(x,y) \in U_3(\vec{G},\mathcal{V}) \cap U(\vec{G}_i', \mathcal{V}_i)$. By definition of $U(\vec{G}_i', \mathcal{V}_i)$, this implies $x, y \in V_i$. Moreover, by definition of $U_3(\vec{G},\mathcal{V})$, we have $(x,y) \notin E$, which immediately implies $(x,y) \in E(\vec{G}_i')$. By Lemma 5.6, we therefore conclude $(x,y) \in U_1(\vec{G}_i', \mathcal{V}_i)$. Let $V_{i,j}$ be the set of the partition $\mathcal{V}_i$ which contains x. Then, by definition of $U_1(\vec{G}_i', \mathcal{V}_i)$, the color of y is contained in both $V_{i,j}$ and $V_i \setminus V_{i,j}$, i.e., V_i contains at least two vertices of color $\sigma(y)$. However, $(x,y) \in U_3(\vec{G},\mathcal{V})$ and $y \in V_i$ together imply that y is the only vertex of its color in V_i; a contradiction. □

We are now in the position to prove the more general

Lemma 5.8. *All edit sets $U(\vec{G}^*[V'],\mathcal{V})$ constructed in Alg. 5 are pairwise disjoint.*

Proof. First note that, by Lemma 5.6, we have $\sigma(x) \neq \sigma(y)$ for all $(x,y) \in U(\vec{G}^*[V'],\mathcal{V})$. Hence, the digraph $(\vec{G}^*, \sigma)$ remains properly colored during the whole recursion. Moreover, recursive calls on a set V' with $|V'| = 1$ trivially contribute with a UR-cost of zero.

By construction, the partitions in consecutive calls of `Edit()` form a hierarchical refinement such that in each recursive call a single element of $V_i \in \mathcal{V}$ is refined. Clearly edit sets encountered in independent branches of the recursion tree are disjoint because they pertain to disjoint vertex sets. For directly consecutive calls of `Edit()`, Lemma 5.7 states that the edits sets are disjoint. Now consider two recursive call on V' and V'' with $V'' \subset V'$ that are not directly consecutive. Let $\mathcal{V}'$ and $\mathcal{V}''$, resp., be the partitions chosen for the vertex sets V' and V'' of $\vec{G}'$ and $\vec{G}''$ at the beginning of the two recursion steps. We can apply the same arguments as in the proof of Lemma 5.7 to conclude that $U_i(\vec{G}'[V'],\mathcal{V}') \cap U(\vec{G}''[V''],\mathcal{V}'') = \emptyset$, $i \in \{1,2\}$. Finally, assume, for contradiction, that $(x,y) \in U_3(\vec{G}'[V'],\mathcal{V}') \cap U(\vec{G}''[V''],\mathcal{V}'')$. By definition of $U(\vec{G}''[V''],\mathcal{V}'')$, this implies $x,y \in V''$. Moreover, by definition of $U_3(\vec{G}'[V'],\mathcal{V}')$, we have $(x,y) \notin E(\vec{G}')$, which immediately implies $(x,y) \in E(\vec{G}' \triangle U(\vec{G}'[V'],\mathcal{V}'))$, i.e., (x,y) is an arc after the editing in this step. Since both x,y are contained in V'', it follows from Lemma 5.6 that all edit steps on the way from $\vec{G}'[V'']$ to $\vec{G}''[V'']$ must be performed by the set U_3, i.e., they exclusively correspond to arc insertions. Therefore, (x,y) is still an arc in $\vec{G}''[V'']$. By Lemma 5.6, we therefore conclude that $(x,y) \in U_1(\vec{G}''[V''],\mathcal{V}'')$. Let V_x be the set of the partition $\mathcal{V}''$ that contains x. Then, by definition of $U_1(\vec{G}''[V''],\mathcal{V}'')$, the color of y is contained in both V_x and $V'' \setminus V_x$, i.e., V'' contains at least two vertices of color $\sigma(y)$. However, $(x,y) \in U_3(\vec{G}'[V'],\mathcal{V}')$ and $y \in V'' \subset V_{x,y}$ for some $V_{x,y} \in \mathcal{V}'$ together imply that y is the only vertex of its color in V''; a contradiction. □

As an immediate consequence of Lemma 5.8, we have

Corollary 5.8. *The edit cost of Alg. 5 is the sum of the UR-costs $c(\vec{G}^*[V'],\mathcal{V})$ in each recursion step.*

It is important to note that the edits $U(\vec{G}^*[V'],\mathcal{V})$ must be applied immediately in each step (cf. Line 5 in Alg. 5). In particular, Lemma 5.8 and Cor. 5.8 pertain to the partitioning of the edited digraph $(\vec{G}^*, \sigma)$, not to the original digraph $(\vec{G}, \sigma)$.

Theorem 5.6. *Every pair of edited digraph $(\vec{G}^*, \sigma)$ and tree T produced as output by Alg. 5 satisfies $(\vec{G}^*, \sigma) = \vec{G}(T, \sigma)$. In particular, $(\vec{G}^*, \sigma)$ is a BMG.*

Proof. By construction, the tree T is phylogenetic and there is a one-to-one correspondence between the vertices $u \in V(T)$ and the recursion steps, which operate on the sets $V' = L(T(u))$. If $|V'| \geq 2$ (or, equivalently, u is an inner vertex of T), we furthermore have $\mathcal{V} = \{L(T(v)) \mid v \in \mathsf{child}_T(u)\}$ for the partition $\mathcal{V}$ of V' chosen in that recursion step. In the following, we denote by $(\vec{G}^*, \sigma)$ the digraph during the editing process, and by $(\vec{G}, \sigma)$ the input digraph, i.e., as in Alg. 5. For brevity, we write E^* for the arc set of the final edited digraph and $E^T := E(\vec{G}(T, \sigma))$.

Let us assume, for contradiction, that there exists (a) $(x, y) \in E^* \setminus E^T \neq \emptyset$, or (b) $(x, y) \in E^T \setminus E^* \neq \emptyset$. In either case, we set $u := \mathrm{lca}_T(x, y)$ and consider the recursion step on $V' := L(T(u))$ with the corresponding partition $\mathcal{V} := \{L(T(v)) \mid v \in \mathsf{child}_T(u)\}$ chosen for V'. Note that $x \neq y$, and thus $u \in V^0(T)$. Moreover, let v_x be the child of u such that $x \preceq_T v_x$, and $V_x := L(T(v_x)) \in \mathcal{V}$.

Case (a): $(x, y) \in E^* \setminus E^T \neq \emptyset$.
Since $(x, y) \notin E^T$ and by the definition of best matches, there must be a vertex $y' \in V_x$ of color $\sigma(y)$ such that $\mathrm{lca}_T(x, y') \prec_T \mathrm{lca}_T(x, y) = u$, and thus $\sigma(y) \in \sigma(V_x)$. Moreover, we have $V_x \in \mathcal{V}$, $x \in V_x$ and $y \in V' \setminus V_x$. Two subcases need to be considered, depending on whether or not (x, y) is an arc in $\vec{G}^*$ at the beginning of the recursion step. In the first case, the arguments above imply that $(x, y) \in U_1(\vec{G}^*[V'], \mathcal{V})$, and thus, $(x, y) \in U(\vec{G}^*[V'], \mathcal{V})$ by Lemma 5.6. Hence, we delete the arc (x, y) in this step. In the second case, it is an easy task to verify that none of the definitions of $U_1(\vec{G}^*[V'], \mathcal{V})$, $U_2(\vec{G}^*[V'], \mathcal{V})$, and $U_3(\vec{G}^*[V'], \mathcal{V})$ matches for (x, y). Since this step is clearly the last one in the recursion hierarchy that can affect the (non-)arc (x, y), it follows for both subcases that $(x, y) \notin E^*$; a contradiction.

Case (b): $(x, y) \in E^T \setminus E^* \neq \emptyset$.
Since $(x, y) \in E^T$ and by the definition of best matches, there cannot be a vertex $y' \in V_x$ of color $\sigma(y)$ such that $\mathrm{lca}_T(x, y') \prec_T \mathrm{lca}_T(x, y) = u$, and thus $\sigma(y) \notin \sigma(V_x)$. Moreover, we have $V_x \in \mathcal{V}$, $x \in V_x$ and $y \in V' \setminus V_x$. Again, two subcases need to be distinguished depending on whether or not (x, y) is an arc in $\vec{G}^*$ at the beginning of the recursion step. In the first case, the arguments above make it easy to verify that none of the definitions of $U_1(\vec{G}^*[V'], \mathcal{V})$, $U_2(\vec{G}^*[V'], \mathcal{V})$, and $U_3(\vec{G}^*[V'], \mathcal{V})$ matches for (x, y). In the second case, we obtain $(x, y) \in U_2(\vec{G}^*[V'], \mathcal{V})$, and thus, $(x, y) \in U(\vec{G}^*[V'], \mathcal{V})$ by Lemma 5.6. Hence, we insert the arc (x, y) in this step. As before, the (non-)arc (x, y) remains unaffected in any deeper recursion step. Therefore, we have $(x, y) \in E^*$ in both subcases; a contradiction.

Finally, $(\vec{G}^*, \sigma) = \vec{G}(T, \sigma)$ immediately implies that $(\vec{G}^*, \sigma)$ is a BMG. □

Cor. 5.8 suggests a greedy-like "local" approach. In each step, the partition $\mathcal{V}$ is chosen to minimize the score $c(\vec{G}, \mathcal{V})$ in Line 4. The example in Fig. 33 shows, however, that the greedy-like choice of $\mathcal{V}$ does not necessarily yield a globally optimal edit set.

In order to identify arcs that must be contained in every edit set, we first clarify the relationship between the partitions $\mathfrak{P}_{\geq 2}$ on V and the partitions defined by the phylogenetic trees on V.

Lemma 5.9. *Let V be a set with $|V| \geq 2$. Let $\mathfrak{P}_{\geq 2}$ be the set of all partitions $\mathcal{V}$ of V with $|\mathcal{V}| \geq 2$. Then the set $\mathcal{T}_V$ of all phylogenetic trees with leaf set V satisfies $\mathcal{T}_V = \bigcup_{\mathcal{V} \in \mathfrak{P}_{\geq 2}} \mathcal{T}(\mathcal{V})$.*

Proof. For every $\mathcal{V} \in \mathfrak{P}_{\geq 2}$, $\mathcal{T}(\mathcal{V})$ is a set of phylogenetic trees on V. Hence, we conclude $\bigcup_{\mathcal{V} \in \mathfrak{P}_{\geq 2}} \mathcal{T}(\mathcal{V}) \subseteq \mathcal{T}_V$. Conversely, assume that $T \in \mathcal{T}_V$. Since T (with root ρ_T) is a phylogenetic tree and has at least two leaves, we have $|\mathsf{child}_T(\rho_T)| \geq 2$. Together with $L(T(\rho_T)) = L(T) = V$, this implies $\mathcal{V}^* := \{L(T(v)) \mid v \in \mathsf{child}_T(\rho_T)\} \in \mathfrak{P}_{\geq 2}$. In particular, T satisfies $T \in \mathcal{T}(\mathcal{V}^*)$ for some $\mathcal{V}^* \in \mathfrak{P}_{\geq 2}$, and is therefore contained in $\bigcup_{\mathcal{V} \in \mathfrak{P}_{\geq 2}} \mathcal{T}(\mathcal{V})$. □

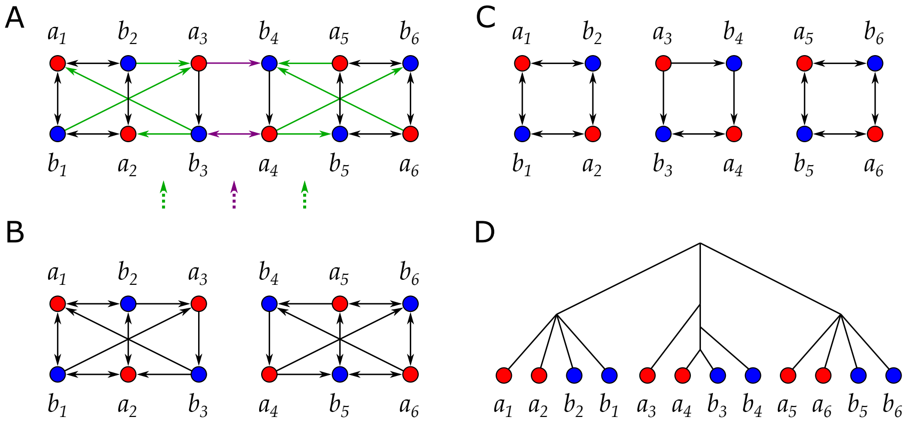

A
C
a_1 b_2 a_3 b_4 a_5 b_6
b_1 a_2 b_3 a_4 b_5 a_6
B
D
a_1 b_2 a_3 b_4 a_5 b_6
b_1 a_2 b_3 a_4 b_5 a_6
a_1 a_2 b_2 b_1 a_3 a_4 b_3 b_4 a_5 a_6 b_5 b_6

Fig. 33. (A) Example for a colored digraph $(\vec{G},\sigma)$ in which the "locally" optimal (first) split does not result in a global optimal BMG editing. The minimal *UR*-cost equals 3 and is attained only for the partition $\mathcal{V} = \{\{a_1, a_2, a_3, b_1, b_2, b_3\}, \{a_4, a_5, a_6, b_4, b_5, b_6\}\}$, which was verified by full enumeration of all partitions and Lemma 5.6. For this partition, $U(\vec{G},\mathcal{V})$ comprises the three purple arcs. (B) The two (isomorphic) induced subgraphs obtained by applying the locally optimal partition $\mathcal{V}$. Each of them has a (global) optimal BMG editing cost of 4. Therefore, the overall symmetric difference of an edited digraph (using the initial split $\mathcal{V}$ as specified) comprises at least $c(\vec{G},\mathcal{V}) + 2 \cdot 4 = 11$ arcs. (C) An optimal editing removes the 8 green arcs and results in a digraph that is explained by the tree in (D). The optimality of this solution was verified using an implementation of the ILP formulation for 2-BMG editing given in Sec. 5.4.1.

Using Lemma 5.9 and given that $|V| \geq 2$, we can express the set of relations that are unsatisfiable for every partition as follows

$$\begin{aligned}\bigcap_{\mathcal{V}\in\mathfrak{P}_{\geq 2}} U(\vec{G},\mathcal{V}) &= \bigcap_{\mathcal{V}\in\mathfrak{P}_{\geq 2}} \left(\bigcap_{T\in\mathfrak{T}(\mathcal{V})} U(\vec{G},T) \right) = \bigcap_{T\in\bigcup_{\mathcal{V}\in\mathfrak{P}_{\geq 2}}\mathfrak{T}(\mathcal{V})} U(\vec{G},T) \\ &= \bigcap_{T\in\mathfrak{T}_V} U(\vec{G},T) = U^*(\vec{G})\,,\end{aligned} \tag{35}$$

i.e., it coincides with the set of relations that are unsatisfiable for every phylogenetic tree, and thus part of every edit set. Note that $U^*(\vec{G})$ is trivially empty if $|V| < 2$. We next show that $U^*(\vec{G})$ can be computed without considering the partitions of V explicitly.

Theorem 5.7. *Let* $(\vec{G} = (V,E),\sigma)$ *be a properly vertex-colored digraph with* $|V| \geq 2$ *then*

$$U^*(\vec{G}) = \{(x,y) \mid (x,y) \notin E,\ x \neq y,\ V[\sigma(y)] = \{y\}\}\,. \tag{36}$$

Proof. First note that $|V| \geq 2$ ensures that $\mathfrak{P}_{\geq 2} \neq \emptyset$. Moreover, since $|\mathcal{V}| \geq 2$ for any $\mathcal{V} \in \mathfrak{P}_{\geq 2}$, the sets $\mathfrak{T}(\mathcal{V})$ are all non-empty as well. With the abbreviation $\hat{U}(\vec{G})$ for the right-hand side of Eq. (36), we show that $\hat{U}(\vec{G}) = \bigcap_{\mathcal{V}\in\mathfrak{P}_{\geq 2}} U(\vec{G},\mathcal{V})$ which by Eq. (35) equals $U^*(\vec{G})$.

Suppose first that $(x,y) \in \hat{U}(\vec{G})$. Then $x \neq y$ and $V[\sigma(y)] = \{y\}$ imply that $\sigma(x) \neq \sigma(y)$. This together with the facts that (i) y is the only vertex of its color in V, and (ii) $L(T) = V$ for each $T \in \mathfrak{T}(\mathcal{V})$ and any $\mathcal{V} \in \mathfrak{P}_{\geq 2}$ implies that y is a best match of x in every such tree T, i.e. $(x,y) \in E(\vec{G}(T,\sigma))$. Since in addition $(x,y) \notin E$ by assumption, we conclude that $(x,y) \in U^*(\vec{G})$.

Now suppose that $(x,y) \in U^*(\vec{G})$. Observe that $\sigma(x) \neq \sigma(y)$ (and thus $x \neq y$) as a consequence of Def. 5.5 and the fact that $(\vec{G},\sigma)$ and all BMGs are properly colored. If $V = \{x,y\}$ and thus $\{\{x\},\{y\}\}$ is the only partition in $\mathfrak{P}_{\geq 2}$, the corresponding unique tree T consists of x and y connected to the root. In this case, we clearly have $(x,y) \in E(\vec{G}(T,\sigma))$ since $\sigma(x) \neq \sigma(y)$. On the other hand, if $\{x,y\} \subsetneq V$, then we can find a partition $\mathcal{V} \in \mathfrak{P}_{\geq 2}$ such that $V_i = \{x,y\}$ for some $V_i \in \mathcal{V}$. In this case, every tree $T \in \mathcal{T}(\mathcal{V})$ has a vertex $v_i \in \mathsf{child}_T(\rho_T)$ with the leaves x and y as its single two children. Clearly, $(x,y) \in E(\vec{G}(T,\sigma))$ holds for any such tree. In summary, there always exists a partition $\mathcal{V} \in \mathfrak{P}_{\geq 2}$ such that $(x,y) \in E(\vec{G}(T,\sigma))$ for some tree $T \in \mathcal{T}(\mathcal{V})$. Therefore, by $(x,y) \in \bigcap_{\mathcal{V}\in\mathfrak{P}_{\geq 2}} U(\vec{G},\mathcal{V})$ and Def. 5.5, we conclude that $(x,y) \notin E$. In order to obtain $(x,y) \in \hat{U}(\vec{G})$, it remains to show that $V[\sigma(y)] = \{y\}$. Since $(x,y) \notin E$ and $(x,y) \in \bigcap_{\mathcal{V}\in\mathfrak{P}_{\geq 2}} U(\vec{G},\mathcal{V})$, it must hold that $(x,y) \in E(\vec{G}(T,\sigma))$ for all $T \in \mathcal{T}(\mathcal{V})$ and all $\mathcal{V} \in \mathfrak{P}_{\geq 2}$. Now assume, for contradiction, that there is a vertex $y' \neq y$ of color $\sigma(y') = \sigma(y)$. Since $\sigma(x) \neq \sigma(y)$, the vertices x, y, y' must be pairwise distinct. Hence, we can find a partition $\mathcal{V} \in \mathfrak{P}_{\geq 2}$ such that $V_i = \{x,y'\}$ for some $V_i \in \mathcal{V}$. In this case, every tree $T \in \mathcal{T}(\mathcal{V})$ has a vertex $v_i \in \mathsf{child}_T(\rho_T)$ with only the leaves x and y' as its children. Clearly, $\mathsf{lca}_T(x,y') = v_i \prec_T \rho_T = \mathsf{lca}_T(x,y)$, and thus $(x,y) \notin E(\vec{G}(T,\sigma))$; a contradiction. Therefore, we conclude that y is the only vertex of its color in V, and hence, $(x,y) \in \hat{U}(\vec{G})$. In summary, therefore, we have $U^*(\vec{G}) = \hat{U}(\vec{G})$. ☐

As a consequence of Thm. 5.7 and by similar arguments as in the proof of Cor. 5.7, we observe

Corollary 5.9. *The set $U^*(\vec{G})$ can be computed in quadratic time.*

By Thm. 5.7, $U^*(\vec{G})$ contains only non-arcs, more precisely, missing arcs pointing towards a vertex that is the only one of its color and thus, by definition, a best match of every other vertex irrespective of the details of the gene tree. By definition, furthermore, $U^*(\vec{G})$ is a subset of every edit set for $(\vec{G},\sigma)$. We therefore have the lower bound

$$|U^*(\vec{G})| \leq c(\vec{G},\mathcal{V}) \tag{37}$$

for every $\mathcal{V} \in \mathfrak{P}_{\geq 2}$.

The following result shows that if $(\vec{G},\sigma)$ is a BMG, then a suitable partition $\mathcal{V}$ can be chosen such that $c(\vec{G},\mathcal{V}) = |U^*(\vec{G})| = 0$.

Lemma 5.10. *Let $(\vec{G} = (V,E),\sigma)$ be a BMG with $|V| \geq 2$ and $\mathcal{V}$ be the connected components of the Aho graph $[\mathcal{R}(\vec{G},\sigma),V]$. Then the partition $\mathcal{V}$ of V satisfies $|\mathcal{V}| \geq 2$ and $c(\vec{G},\mathcal{V}) = 0$.*

Proof. Since $(\vec{G},\sigma)$ is a BMG, we can apply Prop. 4.7 to conclude that $\mathcal{R} := \mathcal{R}(\vec{G},\sigma)$ is consistent and that $(T,\sigma) := (\mathrm{Aho}(\mathcal{R},V),\sigma)$ explains $(\vec{G},\sigma)$, i.e., $\vec{G}(T,\sigma) = (\vec{G},\sigma)$. Hence, $U(\vec{G},T) = \emptyset$. From $|V| \geq 2$ and consistency of $\mathcal{R}$, it follows by Lemma 2.1 that $[\mathcal{R},V]$ has at least two connected components, and thus, by construction, $|\mathcal{V}| \geq 2$. Moreover, we clearly have $T \in \mathcal{T}(\mathcal{V})$ by the construction of T via `BUILD`. Together with $U(\vec{G},T) = \emptyset$, the latter implies $U(\vec{G},\mathcal{V}) = \emptyset$, and thus $c(\vec{G},\mathcal{V}) = 0$. ☐

Lemma 5.11. *Let $(\vec{G} = (V,E),\sigma)$ be a BMG, and $\mathcal{V}$ a partition of V such that $c(\vec{G},\mathcal{V}) = 0$. Then the induced subgraph $(\vec{G}[V'],\sigma_{|V'})$ is a BMG for every $V' \in \mathcal{V}$.*

Proof. Set $\mathcal{R} := \mathcal{R}(\vec{G},\sigma)$ and $\mathcal{F} := \mathcal{F}(\vec{G},\sigma)$ for the sets of informative and forbidden triples of $(\vec{G},\sigma)$, respectively. Since $(\vec{G},\sigma)$ is a BMG, we can apply Prop. 4.7 to conclude that $(\mathcal{R},\mathcal{F})$ is consistent. Now we choose an arbitrary set $V' \in \mathcal{V}$ and set $(\vec{G}',\sigma') := (\vec{G}[V'],\sigma_{|V'})$. By Obs. 4.6, we obtain $\mathcal{R}(\vec{G}',\sigma') = \mathcal{R}_{|V'}$ and $\mathcal{F}(\vec{G}',\sigma') = \mathcal{F}_{|V'}$. This together with the fact that $\mathcal{R}_{|V'} \subseteq \mathcal{R}$ and $\mathcal{F}_{|V'} \subseteq \mathcal{F}$ and Obs. 2.2 implies that $(\mathcal{R}_{|V'},\mathcal{F}_{|V'}) = (\mathcal{R}(\vec{G}',\sigma'),\mathcal{F}(\vec{G}',\sigma'))$ is consistent.

By Prop. 4.7, it remains to show that $(\vec{G}',\sigma')$ is sf-colored to prove that it is a BMG. To this end, assume for contradiction that there is a vertex $x \in V'$ and a color $s \in \sigma(V')$ such that x has no out-neighbor of color $s \neq \sigma(x)$ in V'. However, since the color s is contained in $\sigma(V)$ and $(\vec{G},\sigma)$ is a BMG, and thus sf-colored, we conclude that there must be a vertex $y \in V \setminus V'$ of color s such that $(x,y) \in E$. In summary, we obtain $(x,y) \in E$, $x \in V'$, $y \in V \setminus V'$ and $\sigma(y) = s \in \sigma(V')$. Thus, we have $(x,y) \in U_1(\vec{G},\mathcal{V})$. Hence, Lemma 5.6 implies that $(x,y) \in U(\vec{G},\mathcal{V})$ and, hence, $c(\vec{G},\mathcal{V}) > 0$; a contradiction. Therefore, $(\vec{G}',\sigma')$ must be sf-colored, which concludes the proof. □

Lemma 5.10 and 5.11 allow us to choose the partition $\mathcal{V}$ in each step of Alg. 5 in such a way that Alg. 5 is consistent, i.e., BMGs remain unchanged.

Theorem 5.8. *Alg. 5 is consistent if, in each step on V' with $|V'| \geq 2$, the partition $\mathcal{V}$ in Line 4 is chosen according to one of the following rules:*

1. *$\mathcal{V}$ has minimal UR-cost among all possible partitions $\mathcal{V}'$ of V' with $|\mathcal{V}'| \geq 2$.*
2. *If the Aho graph $[\mathcal{R}(\vec{G}^*[V'],\sigma_{|V'}),V']$ is disconnected with the set of connected components $\mathcal{V}_{\text{Aho}}$, and moreover $c(\vec{G}^*[V'],\mathcal{V}_{\text{Aho}}) = 0$, then $\mathcal{V} = \mathcal{V}_{\text{Aho}}$.*

Proof. We have to show that the final edited digraph $(\vec{G}^*,\sigma)$ returned in Line 13 equals the input digraph $(\vec{G} = (V,E),\sigma)$ whenever $(\vec{G},\sigma)$ already is a BMG, i.e., nothing is edited. Thus suppose that $(\vec{G},\sigma)$ is a BMG and first consider the top-level recursion step on V (where initially $\vec{G}^* = \vec{G}$ still holds at Line 1). If $|V| = 1$, neither $(\vec{G},\sigma)$ nor $(\vec{G}^*,\sigma)$ contain any arcs, and thus, the edit cost is trivially zero. Now suppose $|V| \geq 2$. Since $(\vec{G},\sigma)$ is a BMG, Lemma 5.10 guarantees the existence of a partition $\mathcal{V}$ satisfying $c(\vec{G},\mathcal{V}) = 0$, in particular, the connected components $\mathcal{V}_{\text{Aho}}$ of the Aho graph $[\mathcal{R}(\vec{G},\sigma),V]$ form such a partition. Hence, for both rules (1) and (2), we choose a partition $\mathcal{V}$ with (minimal) *UR*-cost $c(\vec{G},\mathcal{V}) = 0$. Now, Lemma 5.11 implies that the induced subgraph $(\vec{G}[V'],\sigma_{|V'})$ is a BMG for every $V' \in \mathcal{V}$. Since we recurse on these subgraphs, we can repeat the arguments above along the recursion hierarchy to conclude that the *UR*-cost $c(\vec{G}^*[V'],\mathcal{V}')$ vanishes in every recursion step. By Cor. 5.8, the total edit cost of Alg. 5 is the sum of the *UR*-costs $c(\vec{G}^*[V'],\mathcal{V}')$ in each recursion step, and thus, also zero. Therefore, we conclude that we still have $(\vec{G}^*,\sigma) = (\vec{G},\sigma)$ in Line 13. □

By Thm. 5.8, Alg. 5 is consistent whenever the choice of $\mathcal{V}$ minimizes the *UR*-cost of $\mathcal{V}$ in each step. We shall see in Sec. 5.5.3 that minimizing $c(\vec{G},\mathcal{V})$ is a difficult optimization problem in general. Therefore, a good heuristic will be required for this step. This, however, may not guarantee consistency of Alg. 5 in general. The second rule in Thm. 5.8 provides a remedy: the Aho graph $[\mathcal{R}(\vec{G}^*[V'],\sigma_{|V'}),V']$ can be computed efficiently. Whenever $[\mathcal{R}(\vec{G}^*[V'],\sigma_{|V'}),V']$ is not connected, the partition $\mathcal{V}_{\text{Aho}}$ defined by the connected components $[\mathcal{R}(\vec{G}^*[V'],\sigma_{|V'}),V']$ is chosen provided it has *UR*-cost zero. This procedure is effectively a generalization of the algorithm `BUILD` using as input the set of informative triples $\mathcal{R}(\vec{G},\sigma)$ of a properly vertex-colored digraph $(\vec{G},\sigma)$. If $(\vec{G},\sigma)$ is already a BMG, then the recursion in Alg. 5 is exactly the same as in `BUILD`: it recurses on the connected components of the Aho graph (cf. Thm. 4.2). We can summarize this discussion as

Corollary 5.10. *$(\vec{G},\sigma)$ is a BMG if and only if, in every step of the* `BUILD` *algorithm operating on $\mathcal{R}(\vec{G},\sigma)_{|V'}$ and V', either $|V'| = 1$, or $c(\vec{G}^*[V'], \mathcal{V}_{\mathrm{Aho}}) = 0$ for the connected component partition $\mathcal{V}_{\mathrm{Aho}}$ of the disconnected Aho graph $[\mathcal{R}(\vec{G}^*[V'], \sigma_{|V'}), V']$.*

For recursion steps in which the Aho graph $[\mathcal{R}(\vec{G}^*[V'], \sigma_{|V'}), V']$ is connected, and possibly also in steps with non-zero *UR*-cost, another (heuristic) rule has to be employed. As a by-product, we obtain an approach for the case that $\mathcal{R}(\vec{G},\sigma)$ is consistent: Following `BUILD` yields the approximation $\vec{G}(\mathrm{Aho}(\mathcal{R}(\vec{G},\sigma), V(\vec{G})), \sigma)$ as a natural choice.

In the remainder of the section, we will turn to special variants of Alg. 5 for the problem of editing a vertex-colored digraph to a beBMG. As for the more general counterpart of the latter, we have seen that the corresponding decision problem ℓ-BMG EBEG is also NP-complete for $\ell \geq 2$ (cf. Cor. 5.6). Since the recursive partitioning in Alg. 5 defines a tree that explains the edited BMG, see Thm. 5.6, it is reasonable to restrict the optimization of $\mathcal{V}$ in Line 4 to bipartitions. The problem still remains hard, however, since the corresponding decision problem (problem BPURC in Sec. 5.5.3) is NP-complete as shown in Thm. 5.11 below.

Recall that, by Thm. 4.4, a properly vertex-colored digraph $(\vec{G},\sigma)$ with vertex set V is a beBMG if and only if (i) $(\vec{G},\sigma)$ is sf-colored, and (ii) the triple set $\mathcal{R}^{\mathrm{B}}(\vec{G},\sigma)$ is consistent, and that moreover, the BMG $(\vec{G},\sigma)$ is explained by every refinement of the BRT $(\mathrm{Aho}(\mathcal{R}^{\mathrm{B}}(\vec{G},\sigma), V), \sigma)$ in this case. Using this characterization, we can apply analogous arguments as in the proof of Lemma 5.10 for $\mathcal{R}^{\mathrm{B}}(\vec{G},\sigma)$ instead of $\mathcal{R}(\vec{G},\sigma)$ to obtain

Corollary 5.11. *Let $(\vec{G} = (V,E), \sigma)$ be a beBMG with $|V| \geq 2$ and $\mathcal{V}$ be the connected components of the Aho graph $[\mathcal{R}^B(\vec{G},\sigma), V]$. Then the partition $\mathcal{V}$ of V satisfies $|\mathcal{V}| \geq 2$ and $c(\vec{G},\mathcal{V}) = 0$.*

Since a beBMG $(\vec{G},\sigma)$ is explained by every refinement of the BRT (cf. Thm. 4.4), we can obtain a slightly more general result:

Lemma 5.12. *Let $(\vec{G} = (V,E), \sigma)$ be a beBMG with $|V| \geq 2$ and $\mathcal{V}$ be the connected components of the Aho graph $[\mathcal{R}^B(\vec{G},\sigma), V]$. Then, every coarse-graining $\mathcal{V}'$ of $\mathcal{V}$ with $|\mathcal{V}'| \geq 2$ satisfies $c(\vec{G},\mathcal{V}') = 0$.*

Proof. First note that $\mathcal{R}^{\mathrm{B}}(\vec{G},\sigma)$ is consistent by Thm. 4.4 since $(\vec{G},\sigma)$ is a beBMG. Therefore, $|V| \geq 2$ implies $|\mathcal{V}| \geq 2$ [7]. For the trivial coarse-graining $\mathcal{V}' = \mathcal{V}$, Cor. 5.11 already implies the statement. Now assume $\mathcal{V}' \neq \mathcal{V}$. Observe that the tree $(T,\sigma) := (\mathrm{Aho}(\mathcal{R}^{\mathrm{B}}(\vec{G},\sigma), V), \sigma)$ exists and explains $(\vec{G},\sigma)$ by Thm. 4.4. Moreover, there is, by construction, a one-to-one correspondence between the children v_i of its root ρ and the elements in $V_i \in \mathcal{V}$ given by $L(T(v_i)) = V_i$. We construct a refinement (tree) T' of T as follows: Whenever we have multiple sets $V_i \in \mathcal{V}$ that are subsets of the same set $V_j \in \mathcal{V}'$, we remove the edges ρv_i to the corresponding vertices $v_i \in \mathrm{child}_T(\rho)$ in T, and collectively connect these v_i to a newly created vertex w_j. These vertices w_j are then reattached to the root ρ. Since $|\mathcal{V}'| \geq 2$ by assumption, the so-constructed tree T' is still phylogenetic. Moreover, it satisfies $\mathcal{V}' = \{L(T'(v)) \mid v \in \mathsf{child}_{T'}(\rho)\}$, and thus, $T' \in \mathcal{T}(\mathcal{V}')$. It is a refinement of T since contraction of the edges ρw_j again yields T. Hence, we can apply Thm. 4.4 to conclude that (T', σ) also explains $(\vec{G}, \sigma)$. It follows immediately that $U(\vec{G}, T') = \emptyset$. The latter together with $T' \in \mathcal{T}(\mathcal{V}')$ implies $U(\vec{G}, \mathcal{V}') = \emptyset$, and thus $c(\vec{G}, \mathcal{V}') = 0$. ☐

We are now in the position to formulate an analogue of Thm. 5.8 for variants of Alg. 5 that aim to edit a properly-colored digraph $(\vec{G}, \sigma)$ to a beBMG.

Theorem 5.9. *Alg. 5 is consistent for beBMGs $(\vec{G}, \sigma)$ if, in each step on V' with $|V'| \geq 2$, a bipartition $\mathcal{V}$ in Line 4 is chosen according to one of the following rules:*

1. *$\mathcal{V}$ has minimal UR-cost among all possible bipartitions $\mathcal{V}'$ of V'.*
2. *If the Aho graph $[\mathcal{R}^B(\vec{G}^*[V'], \sigma_{|V'}), V']$ is disconnected with the set of connected components $\mathcal{V}_{\mathrm{Aho}}$, and moreover $c(\vec{G}^*[V'], \mathcal{V}_{\mathrm{Aho}}) = 0$, then $\mathcal{V}$ is a coarse-graining of $\mathcal{V}_{\mathrm{Aho}}$.*

Proof. We have to show that the final edited digraph $(\vec{G}^*, \sigma)$ returned in Line 13 equals the input digraph $(\vec{G} = (V, E), \sigma)$ whenever $(\vec{G}, \sigma)$ already is a beBMG, i.e., nothing is edited. Thus suppose that $(\vec{G}, \sigma)$ is a beBMG and first consider the top-level recursion step on V (where initially $\vec{G}^* = \vec{G}$ still holds at Line 1). If $|V| = 1$, neither $(\vec{G}, \sigma)$ nor $(\vec{G}^*, \sigma)$ contain any arcs, and thus, the edit cost is trivially zero. Now suppose $|V| \geq 2$. Since $(\vec{G}, \sigma)$ is a beBMG, $\mathcal{R}^B := \mathcal{R}^B(\vec{G}, \sigma)$ is consistent, and thus, the set of connected components $\mathcal{V}_{\mathrm{Aho}}$ of the Aho graph $[\mathcal{R}^B, V]$ has a cardinality of at least two. If $|\mathcal{V}_{\mathrm{Aho}}| = 2$, $\mathcal{V} := \mathcal{V}_{\mathrm{Aho}}$ is a bipartition satisfying $c(\vec{G}, \mathcal{V}) = 0$ by Cor. 5.11. If $|\mathcal{V}_{\mathrm{Aho}}| > 2$, we can find an arbitrary bipartition $\mathcal{V}$ that is a coarsement of $\mathcal{V}_{\mathrm{Aho}}$. By Lemma 5.12, $\mathcal{V}$ also satisfies $c(\vec{G}, \mathcal{V}) = 0$ in this case. Hence, for both rules (1) and (2), we choose a bipartition $\mathcal{V}$ with (minimal) *UR*-cost $c(\vec{G}, \mathcal{V}) = 0$. Now, Lemma 5.11 implies that the induced subgraph $(\vec{G}[V'], \sigma_{|V'})$ is a BMG for every $V' \in \mathcal{V}$. To see that $(\vec{G}[V'], \sigma_{|V'})$ is also binary-explainable, first note that $\mathcal{R}^B(\vec{G}[V'], \sigma_{|V'}) = \mathcal{R}^B{}_{|V'}$ by Obs. 4.6. This together with the fact that $\mathcal{R}^B{}_{|V'} \subseteq \mathcal{R}^B$ and Obs. 2.2 implies that $\mathcal{R}^B(\vec{G}[V'], \sigma_{|V'})$ is consistent. Moreover, Prop. 4.7 and $(\vec{G}[V'], \sigma_{|V'})$ being a BMG together imply that $(\vec{G}[V'], \sigma_{|V'})$ is sf-colored. Hence, we can apply Thm. 4.4 to conclude that $(\vec{G}[V'], \sigma_{|V'})$ is a beBMG.

Since we recurse on the subgraphs $(\vec{G}[V'], \sigma_{|V'})$, which are again beBMGs, we can repeat the arguments above along the recursion hierarchy to conclude that the *UR*-cost $c(\vec{G}^*[V'], \mathcal{V}')$ vanishes in every recursion step. By Cor. 5.8, the total edit cost of Alg. 5 is the sum of the *UR*-costs $c(\vec{G}^*[V'], \mathcal{V}')$ in each recursion step, and thus, also zero. Therefore, we conclude that we still have $(\vec{G}^*, \sigma) = (\vec{G}, \sigma)$ in Line 13. ☐

5.5.3 *Minimizing the UR-Cost $c(\vec{G}, \mathcal{V})$*

The problem of minimizing $c(\vec{G}, \mathcal{V})$ for a given properly colored digraph $(\vec{G}, \sigma)$ corresponds to the following decision problem.

Problem 5.7 ((BI)PARTITION WITH *UR*-COST ((B)PURC)).

Input:	*A properly ℓ-colored digraph $(\vec{G} = (V, E), \sigma)$ and an integer $k \geq 0$.*
Question:	*Is there a (bi)partition $\mathcal{V}$ of V such that $c(\vec{G}, \mathcal{V}) \leq k$?*

We will show now that BPURC is NP-hard by reduction from SET SPLITTING, one of Garey and Johnson's [99] classical NP-complete problems:

Problem 5.8 (Set Splitting).

Input: *A collection $\mathfrak{C}$ of subsets of a finite set S, denoted by $(\mathfrak{C}, S)$.*

Question: *Is there a bipartition of S into two subsets S_1 and S_2 such that no subset in $\mathfrak{C}$ is entirely contained in either S_1 or S_2?*

Theorem 5.10. [198] Set Splitting *is NP-complete.*

Theorem 5.11. BPURC *is NP-complete.*

Proof. Given a properly vertex-colored digraph $(\vec{G} = (V, E), \sigma)$ and a bipartition $\mathcal{V}$ of V, the set $U(\vec{G}, \mathcal{V})$ and thus the *UR*-cost $c(\vec{G}, \mathcal{V}) = |U(\vec{G}, \mathcal{V})|$ can be computed in polynomial time according to Cor. 5.7. Therefore, BPURC is contained in NP. To show NP-hardness, we use reduction from Set Splitting.

Let $(\mathfrak{C}, S)$ be an instance of Set Splitting. We may assume w.l.o.g. that $|C| \geq 2$ holds for all $C \in \mathfrak{C}$, since otherwise there is no solution at all for Set Splitting. In addition, we assume that $\bigcup_{C \in \mathfrak{C}} C = S$. To see that this does not yield a loss of generality, suppose that $\bigcup_{C \in \mathfrak{C}} C = S' \subsetneq S$. If $\{S'_1, S'_2\}$ is a solution for $(\mathfrak{C}, S')$ then no subset in $\mathfrak{C}$ is entirely contained in either S'_1 or S'_2. Therefore, we can construct a solution (S_1, S_2) for $(\mathfrak{C}, S)$ by arbitrarily adding the elements in $S \setminus S'$ to either S'_1 or S'_2. In contrast, $\{S_1 \cap S', S_2 \cap S'\}$ is a solution for $(\mathfrak{C}, S')$ provided that $\{S_1, S_2\}$ is a solution for $(\mathfrak{C}, S)$.

Now, let $(\mathfrak{C}, S)$ be an instance of Set Splitting and define, for all $s \in S$, the set $\mathfrak{C}(s) := \{C \mid C \in \mathfrak{C}, s \in C\}$ as the subset of $\mathfrak{C}$ that comprises all elements $C \in \mathfrak{C}$ that contain s. Note that $\mathfrak{C}(s) \neq \emptyset$ for all $s \in S$, since we have assumed $\bigcup_{C \in \mathfrak{C}} C = S$, i.e., every $s \in S$ is contained in some element of $\mathfrak{C}$.

We construct a digraph $(\vec{G} = (V, E), \sigma)$ that serves as input for BPURC as follows:

Step 1: For all $s \in S$, construct an s-gadget $\mathcal{G}_s$ as follows:

(i) For all $C \in \mathfrak{C}(s)$, add four new vertices to $\mathcal{G}_s$ of which two are colored with $(C, 1)$ and the other two with $(C, 2)$.

(ii) Add arcs $(x, y), (y, x)$ between all $x, y \in V(\mathcal{G}_s)$ with $\sigma(x) \neq \sigma(y)$.

Step 2: Set $V := \bigcupdot_{s \in S} V(\mathcal{G}_s)$, $E := \bigcupdot_{s \in S} E(\mathcal{G}_s)$ and preserve the coloring of the vertices within the s-gadgets to obtain the digraph $(\vec{G}, \sigma)$.

By construction, $|V(\mathcal{G}_s)| = 4|\mathfrak{C}(s)|$ and $|V| = \sum_{s \in S} 4|\mathfrak{C}(s)| \leq 4|\mathfrak{C}||S|$. Hence, the construction of $(\vec{G}, \sigma)$ can be achieved in polynomial time. Moreover, by construction, $\{V(\mathcal{G}_s) \mid s \in S\}$ forms a partition of V and there are no arcs between vertices of distinct s-gadgets. Furthermore, $\sigma(V) = \mathfrak{C} \times \{1, 2\}$. An illustrative example of such a constructed digraph $(\vec{G}, \sigma)$ is provided in Fig. 34.

We continue by showing that an instance $(\mathfrak{C}, S)$ of Set Splitting has a yes-answer if and only if BPURC has a yes-answer for the input digraph $(\vec{G} = (V, E), \sigma)$ constructed above and $k = 0$. In particular, we will show that $\{S_1, S_2\}$ is a solution of $(\mathfrak{C}, S)$ if and only if $\mathcal{V} = \{V_1, V_2\}$ with $V_i = \cup_{s \in S_i} V(\mathcal{G}_s)$, $i \in \{1, 2\}$ is a solution for $(\vec{G}, \sigma)$ where $c(\vec{G}, \mathcal{V}) = 0$.

Recall that the set of unsatisfiable relations $U(\vec{G}, \mathcal{V})$ of a bipartition $\mathcal{V}$ of V is given by the (disjoint) union $U_1 \cupdot U_2 \cupdot U_3$ of the three sets $U_1 := U_1(\vec{G}, \mathcal{V})$, $U_2 := U_2(\vec{G}, \mathcal{V})$ and $U_3 := U_3(\vec{G}, \mathcal{V})$ (cf. Lemma 5.6).

First suppose that Set Splitting with input $(\mathfrak{C}, S)$ has a yes-answer and let $\{S_1, S_2\}$ be one of its solutions. Hence, no subset in $\mathfrak{C}$ is entirely contained in either S_1 or S_2, and both sets must be non-empty. Consider the set $\mathcal{V} = \{V_1, V_2\}$ with $V_i = \cup_{s \in S_i} V(\mathcal{G}_s)$, $i \in \{1, 2\}$. Since $\{S_1, S_2\}$ is a bipartition of S and $\{V(\mathcal{G}_s) \mid s \in S\}$ is a partition of V, we conclude that $\mathcal{V}$ is a bipartition of V and that $V(\mathcal{G}_s)$ is entirely

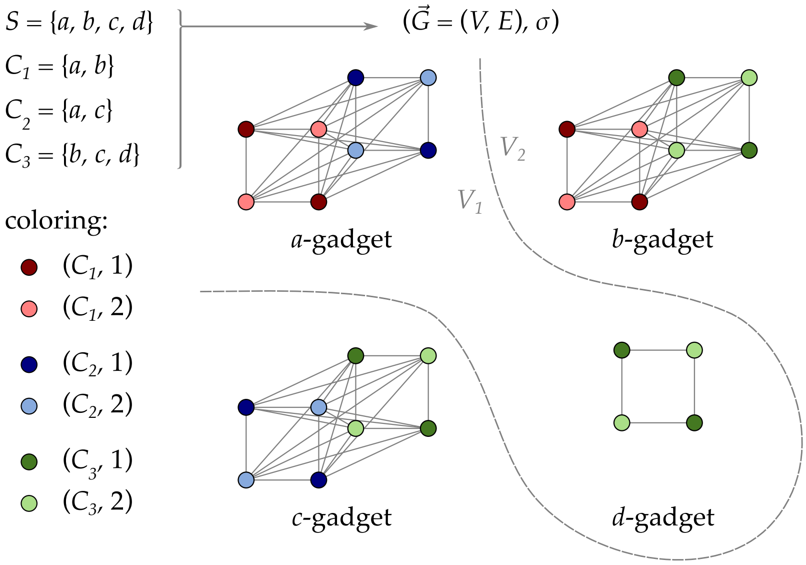


Fig. 34. Example for the reduction from an instance $(\mathfrak{C}, S)$ of Set Splitting to an instance $(\vec{G}, \sigma)$ with $k = 0$ of BPURC, as specified in the proof of Thm. 5.11. In this example, we have $S = \{a, b, c, d\}$ and $\mathfrak{C} = \{C_1, C_2, C_3\}$. By construction, all arcs are bidirectional and thus, arrow heads are omitted in the drawing of $(\vec{G}, \sigma)$. A solution for $(\mathfrak{C}, S)$ is $S_1 = \{a, d\}$ and $S_2 = \{b, c\}$. The latter is equivalent to a solution of BPURC by "separating" the a- and d-gadget from the b- and c-gadget as indicated by the dashed line. The latter yields a bipartition $\mathcal{V} = \{V_1, V_2\}$ of $V(\vec{G})$ that solves BPURC with input $(\vec{G}, \sigma, k = 0)$.
Note, slight changes of the input $(\mathfrak{C}, S)$ to $S' = S \setminus \{d\}$ and $\mathfrak{C}' = \{C_1, C_2, C_3 \setminus \{d\}\}$ would yield an instance of Set Splitting that has no yes-answer. In this case, the d-gadget would disappear from $(\vec{G}, \sigma)$ resulting in the digraph $(\vec{G}', \sigma')$. It is easy to see that there is no bipartition $\mathcal{V} = \{V_1, V_2\}$ of $V(\vec{G}')$ such that $\sigma(V_1) = \sigma(V_2) = \sigma(V(\vec{G}'))$ and no gadget gets split up between V_1 and V_2; two necessary properties to obtain a solution for BPURC with input $(\vec{G}', \sigma')$ and $k = 0$ (cf. proof of Thm. 5.11).

contained in either V_1 or V_2 for all $s \in S$. Together with the fact that there are no arcs in $\vec{G}$ between vertices of distinct s-gadgets this implies that $U_1 = \emptyset$.

In order to verify that $U_2 = U_3 = \emptyset$, we first show that $\sigma(V_1) = \sigma(V_2) = \sigma(V)$ and that V_1 and V_2 contain at least two vertices of every color, respectively. Consider two arbitrary pairs $(C, 1), (C, 2) \in \sigma(V) = \mathfrak{C} \times \{1, 2\}$. Since $\{S_1, S_2\}$ is a solution for Set Splitting with input $(\mathfrak{C}, S)$, there are vertices $s \in C \cap S_1$ and $s' \in C \cap S_2$ and thus, $V(\mathcal{G}_s) \subseteq V_1$ and $V(\mathcal{G}_{s'}) \subseteq V_2$. By construction, each of the sets $V(\mathcal{G}_s)$ and $V(\mathcal{G}_{s'})$ contains two vertices of color $(C, 1)$ and two vertices of color $(C, 2)$. Since $V(\mathcal{G}_s) \subseteq V_1$ and $V(\mathcal{G}_{s'}) \subseteq V_2$, the sets V_1 and V_2 each contain two vertices of both colors $(C, 1)$ and $(C, 2)$. Since $(C, 1), (C, 2) \in \sigma(V)$ are arbitrary and $\sigma(V) = \mathfrak{C} \times \{1, 2\}$, we can conclude that $\sigma(V_1) = \sigma(V_2) = \sigma(V)$, and that V_1 and V_2 contain at least two vertices of every color. Now, $\sigma(V_1) = \sigma(V_2)$ implies that $U_2 = \emptyset$. Moreover, since V_1 and V_2 contain at least two vertices of every color, we also have that $U_3 = \emptyset$. In summary, we have $U(\vec{G}, \mathcal{V}) = U_1 \uplus U_2 \uplus U_3 = \emptyset$, and thus, $c(\vec{G}, \mathcal{V}) = 0$. Therefore, BPURC with input $(\vec{G}, \sigma, k = 0)$ has a yes-answer.

Now suppose BPURC with input $(\vec{G}, \sigma, k = 0)$ has a yes-answer and thus, a solution $\mathcal{V} = \{V_1, V_2\}$. Consequently, $U(\vec{G}, \mathcal{V}) = U_1 \uplus U_2 \uplus U_3 = \emptyset$. We first show that both V_1 and V_2 must contain a vertex of every color in $\sigma(V) = \mathfrak{C} \times \{1, 2\}$. To this end, we assume for contradiction that w.l.o.g. V_1 contains no vertex of color $(C, 1)$ for some $C \in \mathfrak{C}$. Since $|C| \geq 2$, C contains two distinct elements $s, s' \in S$. Note that

$C \in \mathfrak{C}(s)$ and $C \in \mathfrak{C}(s')$. By construction in Step 1, there are vertices $y \in V(\mathcal{G}_s)$ and $y' \in V(\mathcal{G}'_s)$ of color $\sigma(y) = \sigma(y') = (C,1)$. Since $(C,1) \notin \sigma(V_1)$, it must hold that $y, y' \in V_2$. Now consider an arbitrary vertex $x \in V_1$. Note that $(C,1) \notin \sigma(V_1)$ implies $\sigma(x) \neq (C,1)$. Since $\mathcal{G}_s$ and $\mathcal{G}_{s'}$ are, by construction, vertex disjoint, x cannot belong two both gadgets $\mathcal{G}_s$ and $\mathcal{G}_{s'}$. Therefore, we can choose $\tilde{y} \in \{y, y'\}$ such that x and $\tilde{y}$ belong to distinct gadgets, and we obtain $(x, \tilde{y}) \notin E$ by construction. This together with $x \in V_1$, $\tilde{y} \in V_2 = V \setminus V_1$ and $\sigma(\tilde{y}) = (C,1) \notin \sigma(V_1)$ implies $(x, \tilde{y}) \in U_2$. Hence, $U_2 \neq \emptyset$; a contradiction. Therefore, we conclude that both V_1 and V_2 contain vertices of all colors in $\sigma(V) = \mathfrak{C} \times \{1,2\}$.

We continue by showing that $V(\mathcal{G}_s)$ is entirely contained in either V_1 or V_2 for all $s \in S$. To this end, assume for contradiction that there is a gadget $\mathcal{G}_s$ such that $W_1 := V_1 \cap V(\mathcal{G}_s)$ and $W_2 := V_2 \cap V(\mathcal{G}_s)$ are both non-empty. Since $V(\mathcal{G}_s)$ forms a connected component in $(\vec{G}, \sigma)$ and all arcs are bidirectional by construction, we can find two vertices $x \in W_1$ and $y \in W_2$ such that $(x,y) \in E$. This together with the facts that x and y are in distinct sets V_1 and V_2 and that both V_1 and V_2 contain all colors of $\sigma(V)$, implies that $(x,y) \in U_1$. Hence, $U_1 \neq \emptyset$; a contradiction. Therefore, the vertex set of each s-gadget is entirely contained in either V_1 or V_2.

We can construct a well-defined partition $\{S_1, S_2\}$ of S such that $s \in S_i$ if and only if $V(\mathcal{G}_s) \subseteq V_i$, $i \in \{1,2\}$. By construction, there are vertices of color $(C,1)$ and $(C,2)$ in $\mathcal{G}_s$ if and only if $s \in C$. This together with the fact that both V_1 and V_2 contain vertices of all colors $\mathfrak{C} \times \{1,2\}$ implies that $S_1 \cap C$ and $S_2 \cap C$ are both non-empty for every $C \in \mathfrak{C}$. Hence, $\{S_1, S_2\}$ is a solution for Set Splitting with input $(\mathfrak{C}, S)$. ☐

Thm. 5.8 suggests to consider heuristics for (B)PURC that make use of the Aho graph in the following manner:

1. Construct the Aho graph $H := [\mathcal{R}(\vec{G}, \sigma), V]$ based on the set of informative triples $\mathcal{R}(\vec{G}, \sigma)$.
2. If H has more than one connected component, we use the set of connected components as the partition $\mathcal{V}$.
3. If H is connected, a heuristic that operates on the Aho graph H is used to find a partition $\mathcal{V}$ with small *UR*-cost $c(\vec{G}, \mathcal{V})$.

Plugging any algorithm of this type into Line 4 of Alg. 5 reduces the algorithm to `BUILD` if a BMG is used as input and thus guarantees consistency (cf. Thm. 4.2). We note, however, that the connected components of a disconnected Aho graph are not guaranteed to correspond to an optimal solution for (B)PURC in the general case.

5.5.4 *Computational Experiments*

In this section, we compare different heuristics for the (B)PURC Problem and their performance in the context of BMG editing. Somewhat unexpectedly, but in accordance with Fig. 33, the results suggest that a good (or bad) performance of (B)PURC is not directly linked to a good (or bad) performance for BMG editing. Moreover, we find that, even for noisy data, all analyzed methods are able to capture the tree structure of the underlying "true" BMG at least to some extent. As we shall see, community detection approaches in combination with the *UR*-cost appear to be more promising for BMG editing than optimal solutions of (B)PURC alone.

Heuristics for (B)PURC

(B)PURC is a variation on graph partitioning problems. It seems reasonable, therefore, to adapt graph partitioning algorithms for our purposes.

MINCUT. We solve the minimum edge cut problem for the connected undirected graph H, i.e., we want to find a bipartition $\mathcal{V} = \{V_1, V_2\}$ such that the number of edges between V_1 and V_2 is minimal in H. The problem can be solved exactly in polynomial time using the Stoer-Wagner algorithm [300]. Note, however, that the minimum edge cut in H will in general not deliver an optimal solution of (B)PURC.

KARGER'S ALGORITHM is a randomized algorithm that, in its original form, also aims to find a minimum edge cut [163]. In brief, it merges vertices of the graph by randomly choosing and contracting edges, until only two vertices remain, which induce a bipartition $\mathcal{V}$ according to the vertices that were merged into them. By repeating this process a sufficient number of times, a minimum edge cut can be found with high probability. Here, we use the *UR*-cost $c(\vec{G}, \mathcal{V})$ instead of the size of the edge cut as objective function to select the best solution over multiple runs.

A SIMPLE GREEDY APPROACH starts with $\mathcal{V} = \{V_1 = \emptyset, V_2 = V'\}$ and stepwise moves a vertex $v \in V_2$ to V_1 such that $c(\vec{G}, \{V_1 \cup \{v\}, V_2 \setminus \{v\}\})$ is optimized. Ties are broken at random. This produces $|V| - 1$ "locally optimal" bipartitions, from which the best one is selected.

GRADIENT WALKS. Here we interpret the space of all bipartitions $\mathcal{V}$ endowed with the objective function $c(\vec{G}, \mathcal{V})$ as a fitness landscape. We start with a random but balanced bipartition $\mathcal{V} = \{V_1, V_2\}$. As the move set, we allow moving one vertex from V_1 to V_2 or *vice versa*. In each step, we execute the move that best improves the objective function, and stop when we reach a local optimum.

LOUVAIN METHOD. This method for community detection in graphs greedily optimizes the so-called modularity of a vertex partition $\mathcal{V}$ [29]. Its objective function is $q(\mathcal{V}) = \sum_{W \in \mathcal{V}} \sum_{u,v \in W} (a_{uv} - d_u d_v / (2m))$, where a_{uv} are the entries of the (possibly weighted) adjacency matrix of a graph H, $d_u = \sum_v a_{uv}$ the vertex degrees, and m is the sum of all edge weights in the graph. This favors so-called *communities* or *modules* W that are highly connected internally but have only few edges between them. The Louvain method operates in two phases starting from the discrete partition $\mathcal{V} = \{\{u\} \mid u \in V\}$. In the first phase, it repeatedly iterates over all vertices x and moves x into the community of one of its neighbors that leads to the highest gain in modularity as long as a move that increases $q(\mathcal{V})$ can be found. The second phase repeats the first one on the weighted quotient graph $H/\mathcal{V}$ whose vertices are the sets of $\mathcal{V}$ and whose edge weights are the sum of the original weights between the communities. In addition to maximizing the modularity, we also investigate a variant of the Louvain method that moves vertices into the community of one of their neighbors if this results in a lower *UR*-cost $c(\vec{G},\mathcal{V})$, and otherwise proceeds analogously. We exclude the merging of the last two vertices to ensure that a non-trivial partition is returned. Since the Louvain method is sensitive to the order in which the vertex set is traversed, we randomly permute the order of vertices to allow multiple runs on the same input.

With the exception of the Stoer-Wagner algorithm for solving the minimum edge cut problem, all of these partitioning methods include random decisions. One may therefore run them multiple times and use the partition corresponding to the best objective value, i.e., the lowest *UR*-cost $c(\vec{G},\mathcal{V})$ or the highest modularity. If not stated otherwise, we apply five runs for each of these methods in each recursion step (with a connected Aho graph) in the following analyses.

Construction of Test Instances

Since we are interested in the (B)PURC problem in the context of BMG editing, we test the heuristics on ensembles of perturbed BMGs that were constructed as follows: We first generate leaf-colored trees (T,σ) with a predefined number of vertices N and colors ℓ and then compute their BMGs $\vec{G}(T,\sigma)$. For each tree, we start from a single vertex. We then repeatedly choose one of the existing vertices v randomly, and, depending on whether v is currently an inner vertex or a leaf, attach either a single or two new leaves to it, respectively. Hence, the number of leaves increases by exactly one and the tree remains phylogenetic in each step. We stop when the desired number N of leaves is reached. In the next step, colors are assigned randomly to the leaves under the constraint that each of the ℓ colors appears at least once. We note that trees created in this manner are usually not least resolved, and their BMGs are in general not binary-explainable. Finally, we disturb these BMGs by inserting and deleting arcs according to a specified insertion and deletion probability, respectively. Since arcs between vertices of the same color trivially cannot correspond to best matches, we do not insert arcs between such vertices, i.e., the input digraphs for the editing are all properly vertex-colored.

For the purpose of benchmarking the heuristics for the (B)PURC problem, we only retain perturbed BMGs $(\vec{G},\sigma)$ with a connected Aho graph $H := [\mathcal{R}(\vec{G},\sigma), V(\vec{G})]$ because the heuristics are not applied to instances with a disconnected Aho graph H. Depending on the insertion and deletion probabilities, we retained 93% to 100% of the initial sample, except in the case where arcs were only inserted to obtain a disturbed digraph. Here, the Aho graph H was connected in 60% of the initial sample. Thus, even moderate perturbation of a BMG introduces inconsistencies into the triple set $\mathcal{R}(\vec{G},\sigma)$ and results in a connected Aho graph H in the majority of cases. As shown in Fig. 30, both arc insertions and deletions can cause triple inconsistencies.

Benchmarking Results

Fig. 35 suggests that the *Simple Greedy* approach is best suitable for the minimization of the *UR*-cost $c(\vec{G},\mathcal{V})$ for any of the considered parameters for

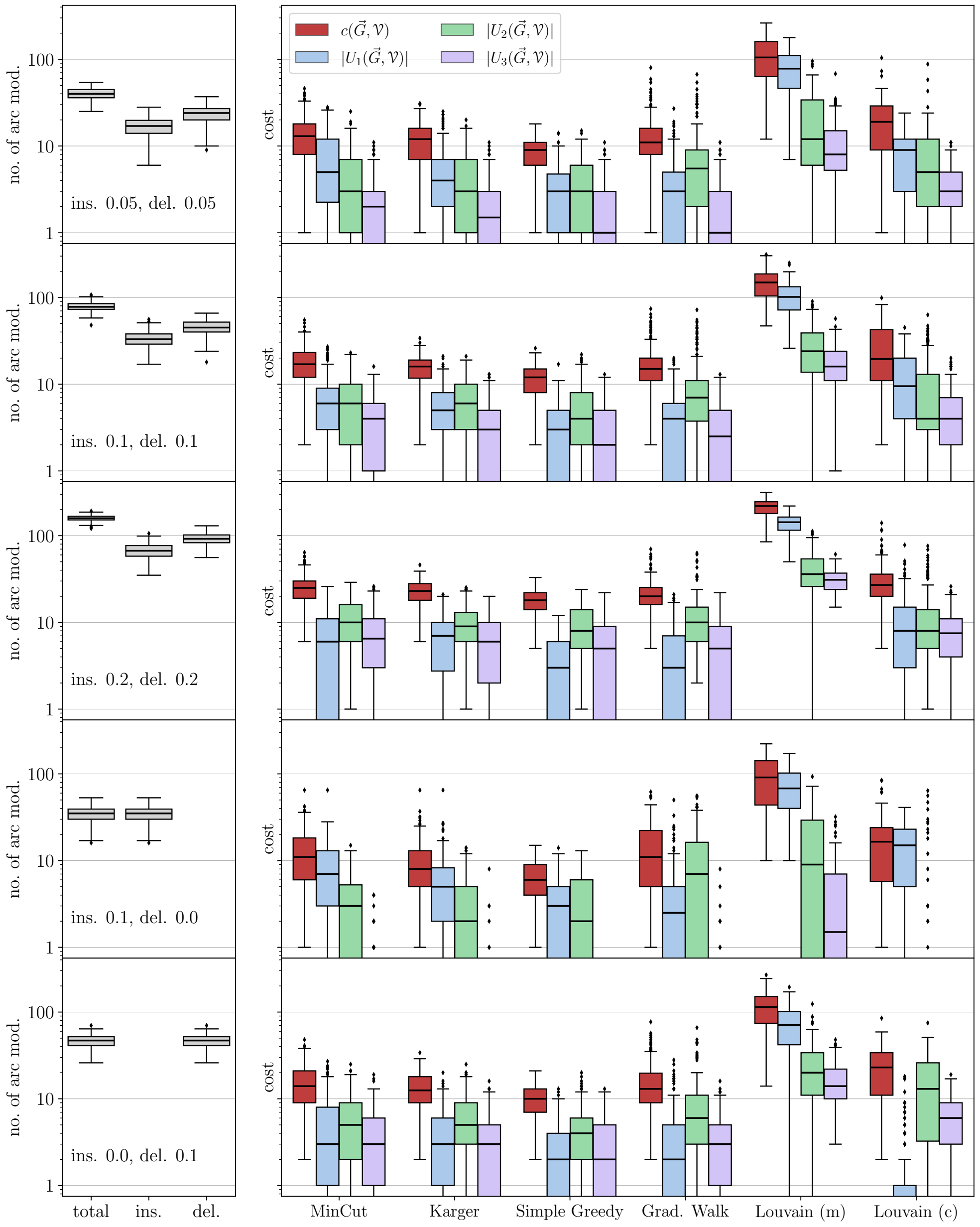


Fig. 35. Performance of partitioning methods for minimizing $c(\vec{G},\mathcal{V})$ on perturbed BMGs $(\vec{G},\sigma)$. The rows correspond to different insertion and deletion probabilities (indicated in the l.h.s. panels) used to disturb the original BMGs. The l.h.s. panels show the distribution of the no. of arc modifications in total, arc insertions and arc deletions of the disturbed digraphs w.r.t. the original BMGs. The r.h.s. panels show the distribution of *UR*-costs $c(\vec{G},\mathcal{V})$ (red) obtained for each method, and of the no. of arcs in $U_1(\vec{G},\mathcal{V})$, $U_2(\vec{G},\mathcal{V})$, and $U_3(\vec{G},\mathcal{V})$ (i.e., the sets that contribute to the *UR*-cost). Example plot for $|V| = 30$ vertices and $|\sigma(V)| = 10$ colors in each digraph. Among the 200 generated digraphs, only those with a connected Aho graph $[\mathcal{R}(\vec{G},\sigma), V(\vec{G})]$ are included in each of the five rows (93%, 100%, 100%, 60%, 95%).

BMG disturbance. The Louvain method based on graph modularity (*Louvain (m)*) appears to have by far the worst performance which, moreover, quickly produces higher *UR*-costs with an increasing intensity of the perturbations.

In order to better understand the behavior of the repeated application of the partitioning heuristics of Alg. 5, it is instructive to consider not only the score but also the structure of partitions. We observe a strong tendency of some of the partitioning methods to produce *single-leaf splits*, i.e., (bi)partitions $\mathcal{V}$ in which at least one set $W \in \mathcal{V}$ is a singleton (i.e., $|W| = 1$). Single-leaf splits in general seem to have relatively low *UR*-costs.

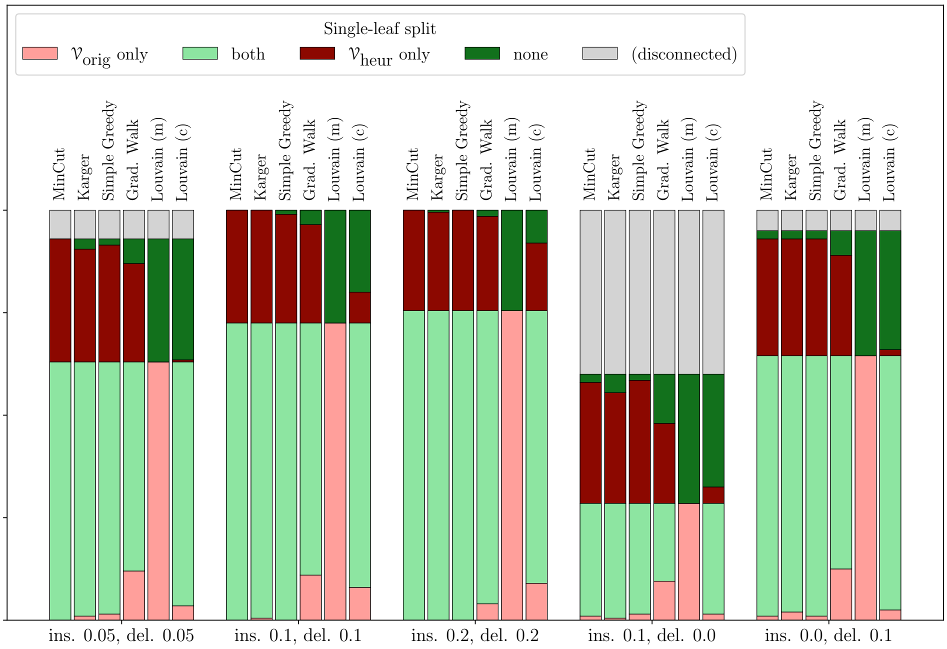


Fig. 36. Abundance of single-leaf splits for pairs of BMGs $(\vec{G}_{\text{orig}}, \sigma)$ and disturbed digraphs $(\vec{G}, \sigma)$ (both with vertex set V). The partition $\mathcal{V}_{\text{orig}}$ corresponds to the connected components of the Aho graph $H_{\text{orig}} := [\mathcal{R}(\vec{G}_{\text{orig}}, \sigma), V]$ and, hence, to the partition induced by the subtrees of the children of the root of the LRT (T, σ) of $(\vec{G}_{\text{orig}}, \sigma)$ (cf. Prop. 4.7). The partition $\mathcal{V}_{\text{heur}}$ corresponds to the partition of V as determined by one of the partitioning methods (based on $H := [\mathcal{R}(\vec{G}, \sigma), V]$). The gray parts of the bars comprise those instances for which H is disconnected. The light and dark red bars indicate the amount of graphs for which only $\mathcal{V}_{\text{orig}}$ or $\mathcal{V}_{\text{heur}}$, resp., is a single-leaf split, while light and dark green bars represent instances for which both and none of the two partitions, resp., are single-leaf splits. Note that the partitions were not compared explicitly, in particular, the identified singletons in $\mathcal{V}_{\text{heur}}$ in the light green instances may deviate from those in $\mathcal{V}_{\text{orig}}$ in some cases. Example plot for $|V| = 30$ vertices and $|\sigma(V)| = 10$ colors in each digraph. 200 generated digraph pairs per combination of arc insertion (ins.) and deletion (del.) probabilities.

Fig. 36 quantifies the abundance of single-leaf splits on the same instances as in Fig. 35. We distinguish between single-leaf splits that are correct w.r.t. the Aho graph H_{orig} of the original unperturbed digraph, and single-leaf splits that are not present in the unperturbed target. MinCut, Karger, Simple Greedy and Gradient Walk frequently produce single-leaf splits that are not present in H_{orig}. The modularity-based Louvain method, in contrast, never returned a single-leaf split, even if it was present in H_{orig}. The modified Louvain method is most often in good agreement with H_{orig} as far as single-leaf splits are concerned, at least for perturbation levels of 10% of insertions and deletions.

Heuristics for BMG Editing

In this section, we explore the performance of several variants of Alg. 4 and 5 for BMG editing. The variants of Alg. 5 correspond to using the heuristics for (B)PURC discussed above for processing a connected Aho graph $H := [\mathcal{R}(\vec{G}^*[V'], \sigma_{|V'}), V']$ for the informative triples $\mathcal{R}(\vec{G}^*[V'], \sigma_{|V'})$ in each step of the recursion. We note that Alg. 5 in combination with any of the heuristics for (B)PURC also serves as a heuristic for MaxRTC because the choice of the partition $\mathcal{V}$ in each recursion step determines a set of included triples $xy|z$, namely those for which x and y are contained in one set of $\mathcal{V}$ while z is contained in another. Another way of expressing that same fact is that an approximation to MaxRTC is given by the subset $\mathcal{R}^* \subseteq \mathcal{R}(\vec{G}, \sigma)$ of the informative triples of the input digraph $(\vec{G}, \sigma)$ that are displayed by the tree T constructed in Alg. 5. In particular, Alg. 5 together with the *MinCut* method has been described as a heuristic for MaxRTC in earlier work [41, 100]. For comparison, we will also consider the following bottom-up approach as a component of Alg. 4:

BEST-PAIR-MERGE-FIRST (BPMF) was described by Wu [333], and constructs a tree from a set of triples $\mathcal{R}$ in a bottom-up fashion. We use here a modified version introduced by Byrka et al. [41]. BPMF operates similar to the well-known UPGMA clustering algorithm [285]. Starting with each vertex $x \in V$ as its own cluster, pairs of clusters are merged iteratively, thereby defining a rooted binary tree with leaf set V. The choice of the two clusters to merge depends on a similarity score with the property that any triple $xy|z$ with x, y, and z lying in distinct clusters S_x, S_y, and S_z contributes positively to $\text{score}(S_x, S_y)$ and negatively to $\text{score}(S_x, S_z)$ and $\text{score}(S_y, S_z)$. Since BPMF constructs the tree T from the bottom, it does not imply a vertex partitioning scheme that could be plugged into the top-down procedure of Alg. 5. Importantly, BPMF is not a consistent heuristic for MaxRTC, i.e. it does not necessarily recognize consistent triples sets. Hence, consistency in the application to BMG editing is also not guaranteed as the example in Fig. 37 shows. Therein, the input digraph $(\vec{G}_{\text{orig}} = (V, E), \sigma)$ is a BMG and explained by $(T_{\text{orig}}, \sigma)$. Therefore, its set of informative triples

$$\begin{aligned}\mathcal{R} :=& \mathcal{R}(\vec{G}_{\text{orig}}, \sigma)\\ =& \{ab_1|b_2,\ ac_1|c_2,\ ac_1|c_3,\ b_1c_1|c_2,\ b_1c_1|c_3,\ b_2c_1|c_2,\ b_2c_1|c_3,\ c_1b_2|b_1\}\end{aligned}$$

is consistent (cf. Prop. 4.7). On the right-hand side of Fig. 37, the first three cluster merging steps in BPMF with input $\mathcal{R}$ are shown where the numbers are the scores $score(S_i, S_j)$ for each pair of clusters S_i and S_j as defined in [41]. The pink arrows link inner vertices of the resulting binary tree (T, σ) and the corresponding cluster merging step based on the maximal score. The tree (T, σ) does not display the triple $ab_1|b_2$. As a consequence, its BMG $\vec{G}(T, \sigma)$ contains the additional arc (a, b_2), and the triple set $\mathcal{R}^*$ extracted from T in

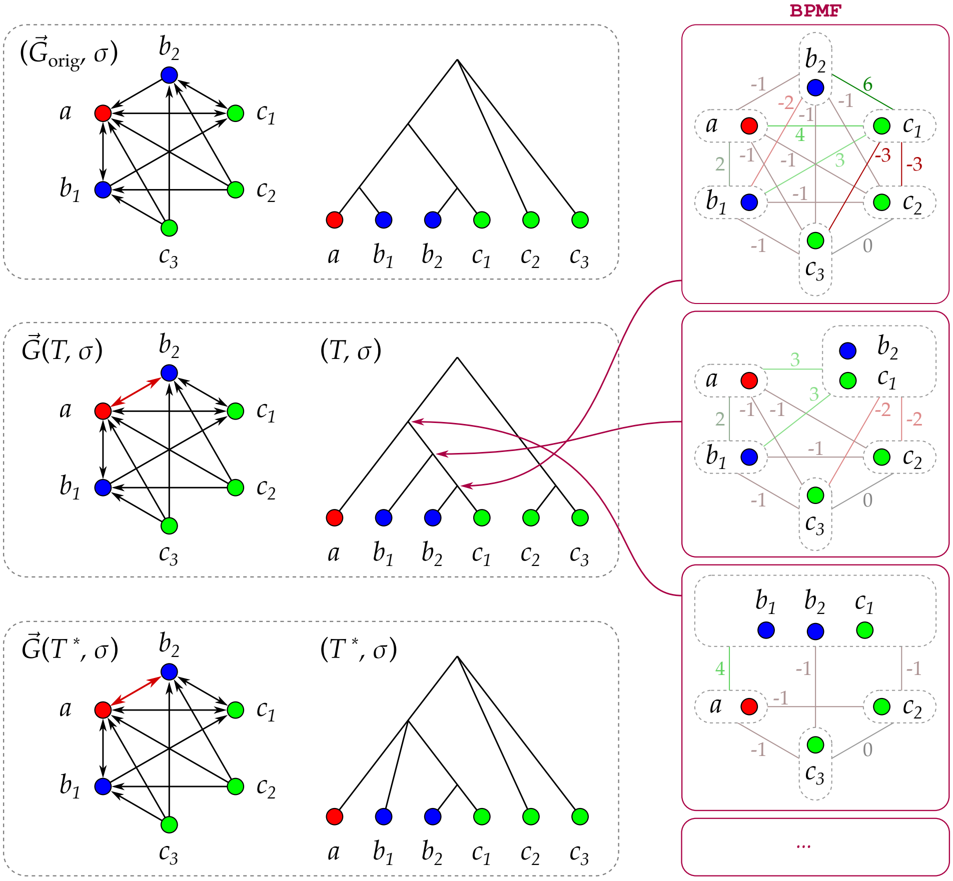


Fig. 37. Example showing that BPMF is not a consistent heuristic for MaxRTC, and that Alg. 4 with BPMF is not a consistent heuristic for BMG editing. See the text for a detailed description.

Alg. 4 is a proper subset of $\mathcal{R}$. In particular, the final editing result $\vec{G}(T^*,\sigma)$ with $T^* = \text{Aho}(\mathcal{R}^*, V)$ also contains the arc (a,b_2) which was not present in the original BMG.

In summary, we have two distinct ways to obtain an edited BMG: We may take either

1. $\vec{G}(T,\sigma)$, where T is the output tree of Alg. 5 or BPMF, respectively, or
2. $\vec{G}(T^*,\sigma)$, where $T^* = \text{Aho}(\mathcal{R}^*, V(\vec{G}))$ is constructed from the consistent triple subset of triples $\mathcal{R}^*$. This corresponds to Alg. 4.

Somewhat surprisingly, the results in Fig. 38 suggest that it is in general beneficial to extract the triple set $\mathcal{R}^*$ and rerun the `BUILD` algorithm, i.e., to use $\vec{G}(T^*,\sigma)$.

Benchmarking Results

To assess the performance of the various heuristics, we consider the differences between the editing result $(\vec{G}^*,\sigma)$ from both the original BMG $(\vec{G}_{orig},\sigma)$ and the perturbed input digraphs $(\vec{G},\sigma)$. In Fig. 38, we summarize the absolute values of the symmetric differences of the arc sets

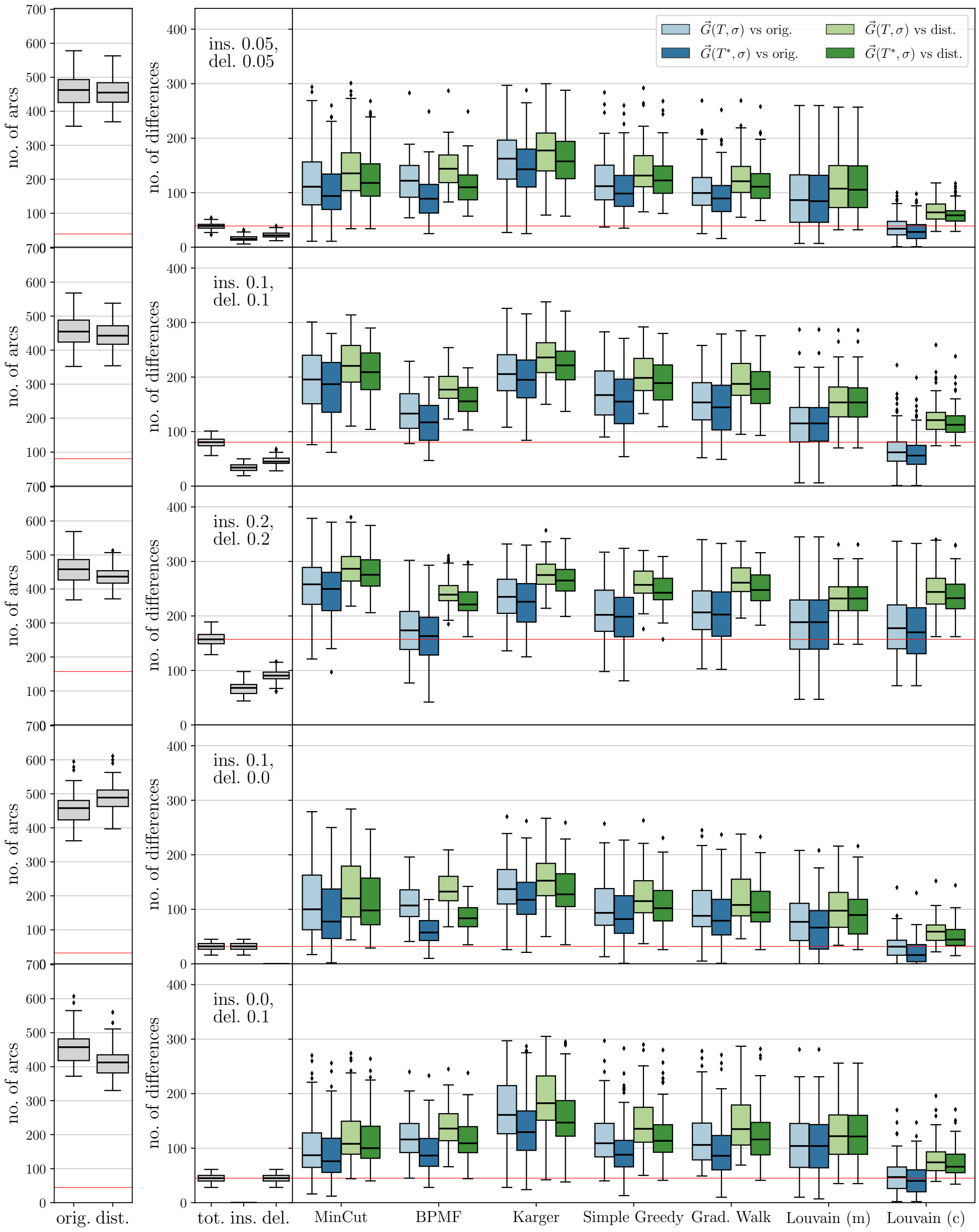


Fig. 38. Performance comparison of several BMG editing heuristics based on the no. of arc differences. The rows correspond to different insertion and deletion probabilities (indicated in the second column panels) used to perturb the original BMGs. The l.h.s. panels show the distribution of the no. of arcs in the original BMG and in the perturbed digraph. The second column panels show the distribution of the no. of arc modifications in total, arc insertions and arc deletions of the perturbed digraphs w.r.t. the original BMGs. The red lines mark the median values of the total no. of modifications. The r.h.s. panels show the total no. of arc differences w.r.t. the original random BMGs (blue) and the perturbed digraphs (green). The light colors indicate the "direct" performance of each method, i.e., the digraph $\vec{G}(T,\sigma)$ where T is the tree that is directly constructed by each method. The darker colors indicate the results if the methods are used as heuristic for MaxRTC in Alg. 4. Example plot for $|V| = 30$ vertices and $|\sigma(V)| = 10$ colors in each digraph, 100 digraphs per row.

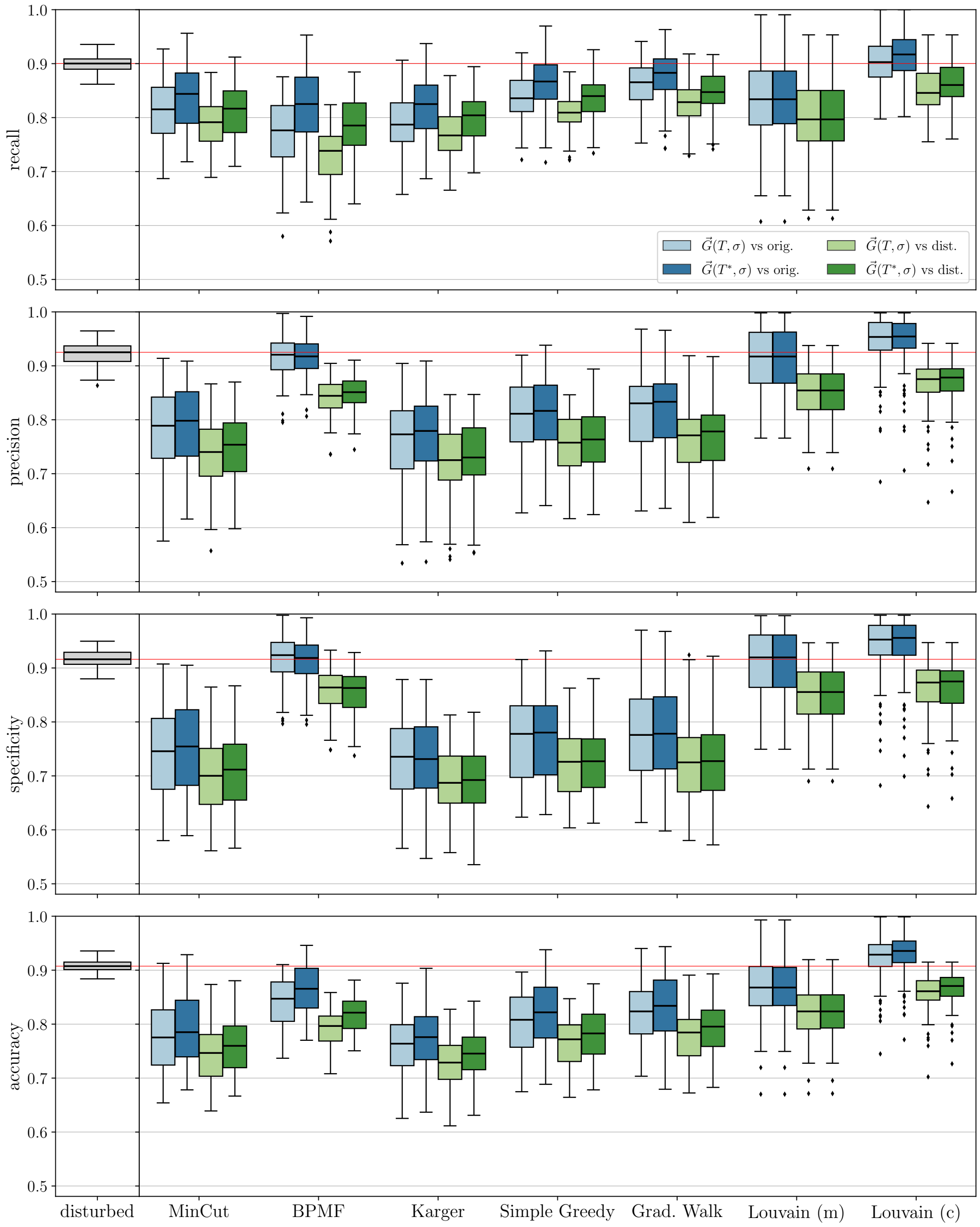


Fig. 39. Performance comparison of several BMG editing heuristics based on recall, precision, specificity, and accuracy (rows 1 to 4). The l.h.s. panels show the respective measure for the perturbed digraph w.r.t. the original random BMG. The red lines marks the median values of the latter. The r.h.s. panels show the results for the edited digraphs w.r.t. the original BMGs (blue) and the perturbed graphs (green). The light colors indicate the "direct" performance of each method, i.e., the digraph $\vec{G}(T,\sigma)$ where T is the tree that is directly constructed by each method. The darker colors indicate the results if the methods are used as heuristic for MaxRTC in Alg. 4. Example plot for $|V| = 30$ vertices and $|\sigma(V)| = 10$ colors in each digraph, insertion and deletion probability 0.1, and 100 digraphs.

$d_{orig} := |E(\vec{G}^*) \triangle E(\vec{G}_{orig})|$ and $d := |E(\vec{G}^*) \triangle E(\vec{G})|$, respectively. These results are translated to usual normalized performance indicators (recall, precision, specificity, and accuracy; all defined in terms of the arc sets) in Fig. 39.

Comparing the distances d_{orig} (blue boxplots) and d (green boxplots) of the editing result $(\vec{G}^*, \sigma)$ to original unperturbed BMG and the input digraph, resp., we find that, for the methods investigated here, on average d_{orig} is smaller than d. This indicates that all methods are able to capture the underlying tree structure of the original BMG at least to some extent. The discrepancy between d_{orig} and d tends to increase with the level of perturbation, a trend that is most pronounced for *Louvain (c)*. This result is encouraging for practical applications of BMG modification to correcting noisy best match data, where the eventual goal is to obtain a good estimate of the underlying true BMG.

Intriguingly, the extraction of consistent informative triples $\mathcal{R}^*$ from the reconstructed tree T and rerunning `BUILD`, i.e., using $\vec{G}(T^*, \sigma)$, in general improves the estimation results for the majority of methods. In particular, this increases the recall without a notable negative impact on precision and specificity (cf. Fig. 39). A better recall, corresponding to a higher proportion of correctly inferred arcs, is not surprising in this context, since this additional step in essence reduces the number of triples. We therefore expect the tree $T^* = \text{Aho}(\mathcal{R}^*, V(\vec{G}))$ to be on average less resolved than T. The BMGs of less resolved trees tend to have more arcs than BMGs of highly resolved tree (cf. [264, Lemma 8]). In good accordance with this prediction, *BPMF*, which shows a strong increase of recall, always constructs a binary, i.e., fully-resolved, tree T – whereas the corresponding tree T^* in general is much less resolved.

Somewhat surprisingly, a good or bad performance for minimizing the *UR*-cost in individual steps apparently does not directly translate to the performance in the overall editing procedure. In particular, the modularity-based *Louvain (m)* method seems to be a better choice than the *Simple Greedy* approach. The methods *MinCut* and *Karger* do not seem to be suitable components for Alg. 5, with the exception of the case where perturbations are arc deletions only (Fig. 38, bottom row). Here, *MinCut* produces reasonable estimates that compare well with other methods. The bottom-up method for the MaxRTC problem *BPMF* also produces relatively good results. It appears to be robust at high levels of perturbation. For most of the parameter combinations, we obtain the best results with the *UR*-cost-based Louvain method (*Louvain (c)*). Here, we often observe a symmetric difference (w.r.t. the arcs sets) that is better than the difference between the original and the perturbed digraph. This trend is illustrated by the red median lines in Fig. 38 and 39. Hence, we achieve two goals of BMG editing: (i) the resulting digraph $(\vec{G}^*, \sigma)$ is a BMG, i.e., it satisfies Def. 4.2, and (ii) it is closer to the original BMG than the perturbed digraph. We note that we observed similar trends across all investigated combinations for the numbers of leaves N (ranging from 10 to 40) and of colors ℓ ($\ell < N$ ranging from 2 to at most 20).

Our results show that minimization of the *UR*-cost in each step is not the best approach to BMG editing because this often produces very unbalanced

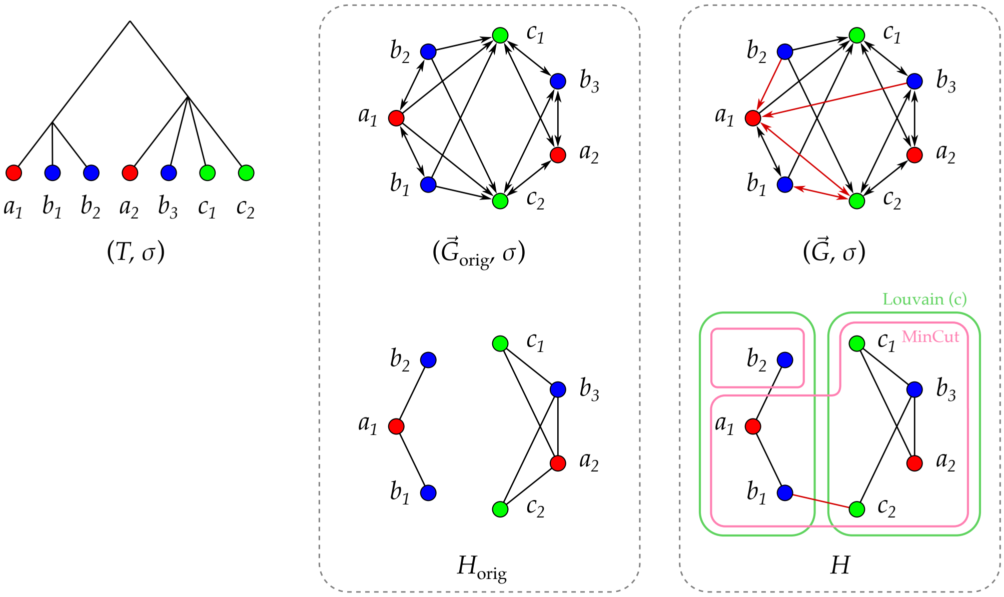

a_1 b_1 b_2 a_2 b_3 c_1 c_2
(T, σ)
$(\vec{G}_{\text{orig}}, \sigma)$
$(\vec{G}, \sigma)$
H_{orig}
H
Louvain (c)
MinCut

Fig. 40. Example of an instance where the Louvain method performs better due to more balanced partitions. The (least resolved) tree (T, σ) explains the BMG $(\vec{G}_{\text{orig}}, \sigma)$ with vertex set V. The graph $H_{\text{orig}} = [\mathcal{R}(\vec{G}_{\text{orig}}, \sigma), V]$ is the Aho graph corresponding to the informative triple set $\mathcal{R}(\vec{G}_{\text{orig}}, \sigma)$. The perturbed digraph $(\vec{G}, \sigma)$ is obtained from $(\vec{G}_{\text{orig}}, \sigma)$ by inserting the arcs (b_3, a_1), (c_2, a_1), and (c_2, b_1) and deletion of (a_1, b_2). The corresponding Aho graph $H = [\mathcal{R}(\vec{G}, \sigma), V]$ is connected because the perturbation introduced the additional informative triple $c_2 b_1 | b_2$. The green and pink frames correspond to the partitions $\mathcal{V}_1$ and $\mathcal{V}_2$ of V constructed by the methods *Louvain (c)* and *MinCut*, respectively.

partitions. As a consequence, more recursion steps are needed in Alg. 5 resulting in higher accumulated number of arc edits. Fig. 40 shows that better solutions to the BMG editing problem are not necessarily composed of vertex partitions with minimal *UR*-cost in each step. The perturbed digraph $(\vec{G}, \sigma)$ in Fig. 40 was obtained from the randomly simulated BMG $(\vec{G}_{\text{orig}}, \sigma)$ as described above using equal insertion and deletion probabilities of 0.1. As an example, the partitions $\mathcal{V}_1$ and $\mathcal{V}_2$ as constructed by the *MinCut* and the *Louvain (c)* method in the first iteration step of Alg. 5 are shown as pink and green frames, respectively. *MinCut* produces a single-leaf split $\mathcal{V}_1$ with an isolated vertex b_2 and *UR*-cost $c(\vec{G}, \mathcal{V}_1) = 1$ deriving from $U_1(\vec{G}, \mathcal{V}_1) = \{(b_2, a_2)\}$. *Louvain (c)* identifies the partition $\mathcal{V}_2$ with $c(\vec{G}, \mathcal{V}_2) = 3$ originating from $U_2(\vec{G}, \mathcal{V}_1) = \{(b_3, a_1), (c_2, a_1), (c_2, b_1)\}$, which corresponds to the connected components of the Aho graph H_{orig} of the unperturbed BMG and thus identifies the split in the original tree (T, σ). Here, the correct partition $\mathcal{V}_2$ has a strictly larger *UR*-cost than the misleading choice of $\mathcal{V}_1$. However, *MinCut* results in a higher total edit cost than *Louvain (c)* for $(\vec{G}, \sigma)$.

In order to account for the issue of unbalanced partitions, we performed a cursory analysis on maximizing a gain function rather than minimizing the *UR*-cost. In analogy to $c(\vec{G}, \mathcal{V})$, we defined $g(\vec{G}, \mathcal{V})$ as the number of arcs and non-arcs that are satisfied by the BMGs of *all* trees in $\mathcal{T}(\mathcal{V})$. Recapitulating the arguments in the proof of Lemma 5.6, one can show that these relations

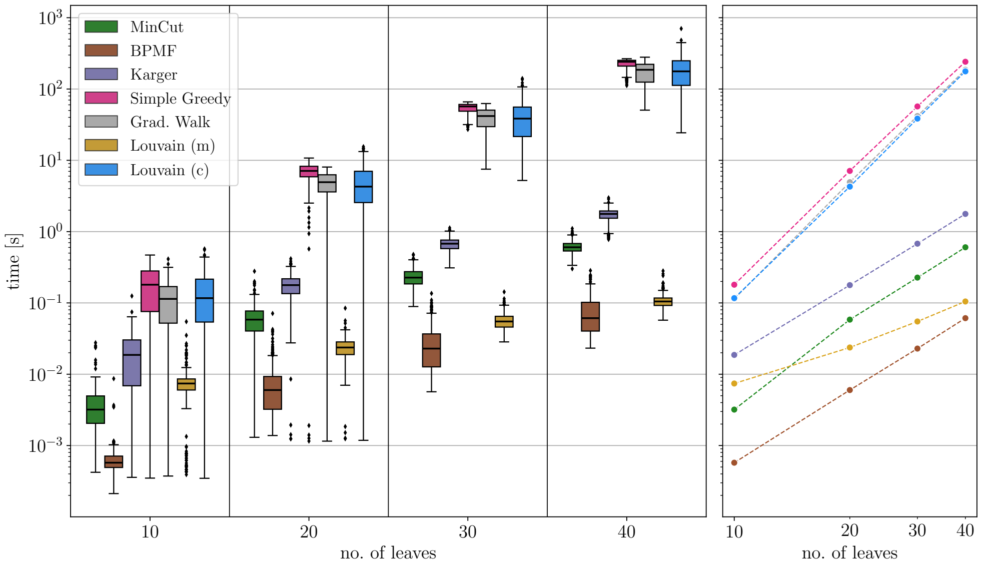


Fig. 41. Running times of the different methods for BMG editing. The time only includes the construction of the tree T, i.e., Alg. 5 or *BPMF*, resp., but not the extraction of the triple set $\mathcal{R}^*$ followed by rerunning `BUILD`. For each number of leaves $N \in \{10, 20, 30, 40\}$ and each number of colors ℓ (taken from $\{2, 5, 10, 20\}$ such that $\ell < N$), 100 perturbed BMGs were generated using equal insertion and deletion probabilities of 0.1. In the right panel, the median values are shown with logarithmic axes.

can also be determined as the union of three sets by replacing "$(x,y) \in E$" with "$(x,y) \notin E$" and *vice versa* in the definitions of $U_1(\vec{G}, \mathcal{V})$, $U_2(\vec{G}, \mathcal{V})$, and $U_3(\vec{G}, \mathcal{V})$. The gain function $g(\vec{G}, \mathcal{V})$ can be used instead of the *UR*-cost with *Karger*, *Simple Greedy*, *Gradient Walk*, and in a gain-function-based *Louvain* method. For all these algorithms, however, maximizing $g(\vec{G}, \mathcal{V})$ leads to partitions that appear to be *too* balanced, and a performance for BMG editing that is worse than the use of the *UR*-cost. A possible explanation for both unbalanced and too balanced partitions as produced with a cost and gain function, resp., is the fact that $U_1(\vec{G}, \mathcal{V})$ and $U_2(\vec{G}, \mathcal{V})$ (and their gain function counterparts) contain pairs of vertices (x,y) that lie in distinct sets of $\mathcal{V}$. Hence, both single-leaf splits and perfectly balanced partitions minimize (maximize, resp.) the number of pairs that could potentially be contained in these arc sets.

All methods for BMG editing were implemented and compared using Python on an off-the-shelf laptop. Fig. 41 summarizes the running times. The right panel shows that all methods appear to scale polynomially in the size $|V|$ of the vertex set of the input digraph. The methods that explicitly rely on the *UR*-cost are much slower than the other methods. We suspect that this is largely due to the repeated $O(|V'|^2)$-computation of $c(\vec{G}, \mathcal{V})$ whenever a vertex is moved between the sets/communities in $\mathcal{V}$. This could possibly be improved by an incremental algorithm.

Heuristics for Binary-Explainable BMG Editing

In order to test the heuristics for the slightly different task of obtaining a binary-explainable BMG $(\vec{G}^*, \sigma)$, we constructed a similar set of test instances. The only difference is that we ensured that T_{orig} is binary by modifying the attachment procedure (cf. Section 5.5.4) such that in each growth step we only choose among the vertices that are currently leaves for attaching two new leaves. Thus, $(\vec{G}_{\text{orig}}, \sigma) = \vec{G}(T_{\text{orig}}, \sigma)$ is binary-explainable. The editing heuristics are analogous, with two straightforward modifications:

- In the Aho graphs, $\mathcal{R}^{\text{B}}(\vec{G}, \sigma)$ is used instead of $\mathcal{R}(\vec{G}, \sigma)$.
- If we encounter a partition $\mathcal{V}$ of cardinality greater than two in some recursion step, we use a coarse-graining $\mathcal{V}'$ of $\mathcal{V}$ such that $|\mathcal{V}'| = 2$ instead. This modification is necessary whenever $[\mathcal{R}^{\text{B}}(\vec{G}, \sigma)[V'], V']$ itself has more than two connected components, and for the partitions with $|\mathcal{V}| \geq 3$ returned by the Louvain method.

By Thm. 5.9, this procedure is consistent for binary-explainable BMGs. Thm. 5.9, moreover, guarantees some freedom in the choice of a coarse-graining $\mathcal{V}' = \{V_1, V_2\}$ whenever $\mathcal{V}$ is not a bipartition. We therefore aim to produce (locally) balanced trees in such situations, i.e., we seek to minimize the difference of $|V_1|$ and $|V_2|$. Formally, this corresponds to the well-known NUMBER PARTITIONING problem with the multiset $\{|V_i| \mid V_i \in \mathcal{V}\}$ as input. We use the efficient heuristic described by Karmarkar and Karp [164], which in general appears to yield very good solutions of the NUMBER PARTITIONING problem [31].

To construct the second binary tree T^* based on subset of triples $\mathcal{R}^* \subseteq \mathcal{R}^{\text{B}}(\vec{G}, \sigma)$ that are displayed by T, we employ an analogous coarse-graining in an otherwise unmodified `BUILD` algorithm. We note, however, that one could incorporate more sophisticated approaches which e.g. use some greedy coarse-graining method based on the *UR*-cost.

The results for beBMG editing in Fig. 42 in essence recapitulate the observations for general BMG editing: Alg. 5 in combination with *Louvain (c)* appears to be the best choice for the majority of parameter combinations. However, it is outperformed by the *BPMF* heuristic at high levels of perturbation (insertion and deletion probability 0.2). As in the general case, construction of T^* and using $(\vec{G}^*, \sigma) := \vec{G}(T^*, \sigma)$ as editing result appears to be advantageous. Moreover, the difference of the editing result $(\vec{G}^*, \sigma)$ to the original beBMG $(\vec{G}_{\text{orig}}, \sigma)$ is on average smaller than the difference of $(\vec{G}^*, \sigma)$ to the perturbed digraph $(\vec{G}, \sigma)$.

Remarks on the Results and Open Problems

For all of the methods investigated here, we found that the Aho graph $H := [\mathcal{R}(\vec{G}, \sigma)[V'], V']$ serves as a useful starting point for finding a suitable partition. This choice is based on the idea that, due to the properties of BMGs and in particular the construction of the tree (T, σ) from informative triples of the BMG $(\vec{G}, \sigma) = \vec{G}(T, \sigma)$, arc insertions and deletions in $(\vec{G}, \sigma)$ should not

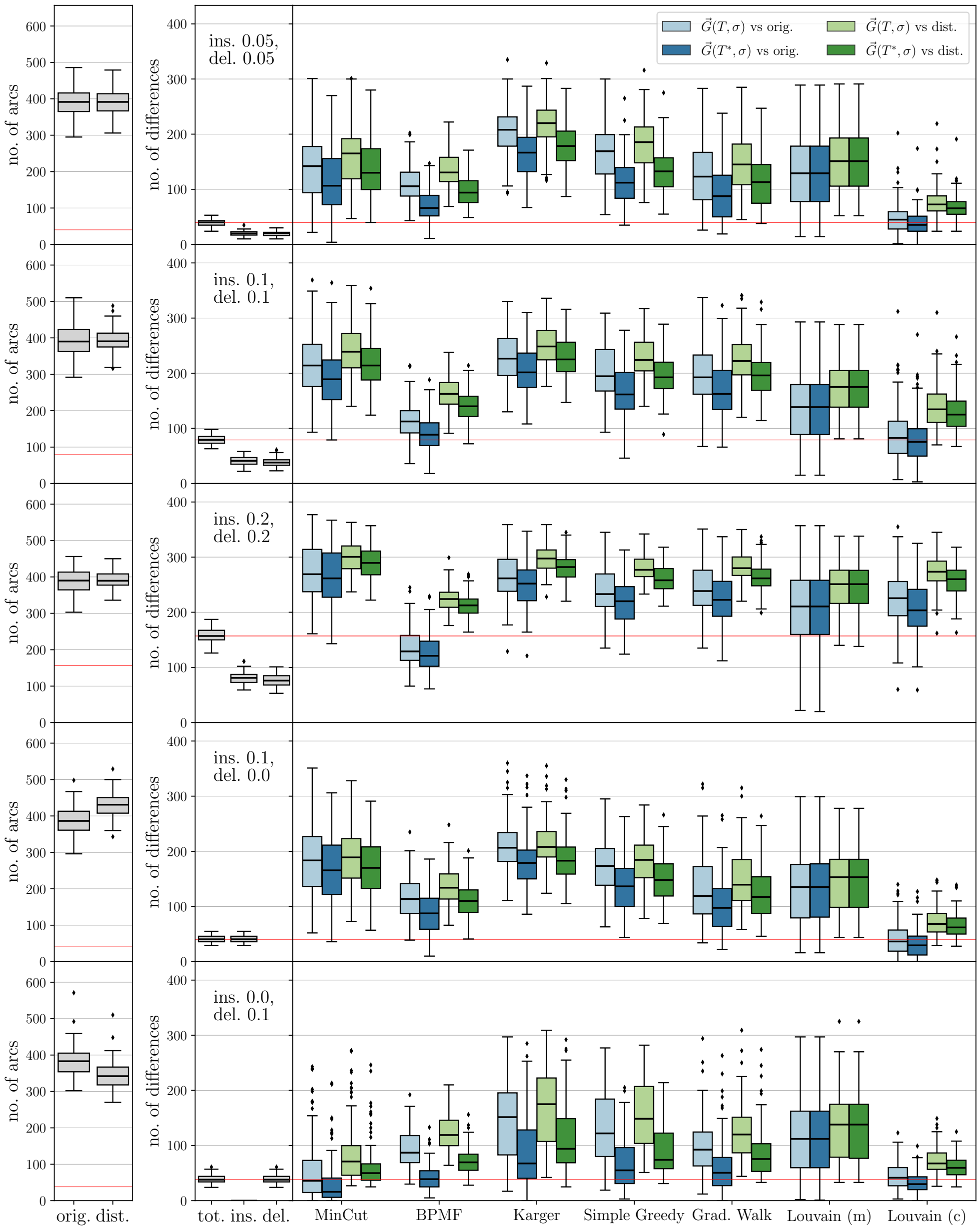


Fig. 42. Performance comparison of several beBMG editing heuristics based on the no. of arc differences. See Fig. 38 for further description. Example plot for $|V| = 30$ vertices and $|\sigma(V)| = 10$ colors in each digraph, 100 digraphs per row.

add too many new edges between the connected components of the originally disconnected Aho graph of $\mathcal{R}(\vec{G},\sigma)$ (cf. Fig. 30). Therefore, we suggest that there is a correlation between good partitions $\mathcal{V}$ of V', i.e. partitions linked to few edits, and the minimization of the number of edges in H connecting vertices in distinct sets of $\mathcal{V}$.

For the general BMG editing problem, we did not make use of the information contained in the set of forbidden triples $\mathcal{F}(\vec{G},\sigma)$ of the input digraph $(\vec{G},\sigma)$. It might be possible to adapt the algorithm `MTT` [128], which identifies consistent pairs $(\mathcal{R},\mathcal{F})$, instead of `BUILD`. `MTT` constructs a coarse-graining

$\mathcal{V}_{\mathrm{MTT}}$ of the set of connected components of the Aho graph (on $\mathcal{R}$) in order to account for the forbidden triples in $\mathcal{F}$ in each recursion step. Possibly, $\mathcal{V}_{\mathrm{MTT}}$ (or some suitable graph representation) yields a further improvement. However, in case of beBMG editing, the extended triple set $\mathcal{R}^{B}(\vec{G}, \sigma)$ and thus the corresponding Aho graphs by construction already cover the information contained in $\mathcal{F}(\vec{G}, \sigma)$. Since no substantial improvement over the general case was observed in this case (cf. Fig. 42), we opted against more detailed benchmarking of $\mathcal{V}_{\mathrm{MTT}}$ in comparison to partitions based on the Aho graph.

We have demonstrated that the problem of BMG editing can be solved for interestingly large instances at reasonable accuracy. In computational biology, however, much larger problems than the ones considered here would also be of interest. Less emphasis has been placed here on computational efficiency and scalability of different variants. We leave this as topic for future research. Given the performance advantage of community detection over minimization of the *UR*-cost in each step, it seems most promising to focus on community detection methods that scale well for very large system. The Louvain method seems to be a promising candidate, since it has been applied successfully to large networks in the past [29]. This is largely due to the fact that the change of modularity in response to moving a vertex between modules can be computed efficiently. We suspect that a comparably fast computation of the *UR*-cost may also be possible; this does not appear to be trivial, however. Moreover, the method could probably be accelerated by moving vertices into the community of the first neighbor such that this results in a (not necessarily optimal) improvement of the *UR*-cost. A similar randomization approach has already shown to only slightly affect the clustering quality in terms of modularity [317].

Since the restriction of a (be)BMG to a subset of colors is again a (be)BMG, it may also be possible to remove large parts of the noise by editing induced subgraph on a moderate number of colors, possibly using information of the phylogeny of the species to select species (= color) sets. Presumably, color sets with sufficient overlaps will need to be considered. A systematic analysis of this idea, however, depends on scalable BMG editing for large instances.

A potential shortcoming of the empirical analysis in this section is the simplistic error model, i.e., the independent perturbation of arcs (and non-arcs). Better models will depend on the investigation of BMGs derived from real-life sequence data. Such data is often burdened with systematic errors arising e.g. from the fact that a common ancestry often cannot be detected for very large evolutionary distances and from unequal mutation rates during the evolution of gene families, see e.g. [185, 250, 293] for more in-depth discussions of these issues. Benchmarking using real-life data, however, is a difficult task because the ground truth is unknown and large, well-curated data sets are not available. As a potential remedy, our results so far suggest that a good performance w.r.t. the input digraph is also an indicator for a good performance w.r.t. the true digraph (cf. Fig. 38 and 39, green vs. blue boxplots).

5.6 SUMMARY

In this chapter, we have investigated the complexity and presented heuristic algorithms for arc set modification problems of an arbitrary vertex-colored digraph $(\vec{G},\sigma)$ to a BMG (or beBMG).

In Secs. 5.2 and 5.3, we have seen that the deletion, completion, and editing problems are all NP-complete (Thm. 5.5 and Cor. 5.6) and that they remain NP-complete even if the input digraph has only two colors, and when the output is additionally constrained to be binary-explainable. However, we were able to derive ILP formulations that are guaranteed to return optimal solutions for all of these problems in Sec. 5.4. Moreover, we gave a simple polynomial-time algorithm for solving the special case of completing a 2-colored BMG to a beBMG (cf. Thm. 5.4).

In Sec. 5.5, we have described a large class of heuristics for BMG editing that operate in a recursive top-down fashion to (at least implicitly) construct a tree (T,σ) capturing the underlying BMG-structure of an arbitrary input digraph $(\vec{G},\sigma)$. We have shown that this is closely related to a specific notion of locally good edits, which we assess using the *UR*-cost. The *UR*-cost counts the minimum number of arc insertions and deletions of the BMG-editing for $(\vec{G},\sigma)$ that are linked to each inner node (and thus to their corresponding leaf partitions) in (T,σ) and cannot be reversed in subsequent recursion steps. In particular, we showed that an optimal solution among all possible partitions guarantees consistency of this class of heuristics (cf. Thm. 5.8 and 5.9). Unfortunately, the corresponding problem BPURC is itself NP-complete (cf. Thm. 5.11).

We therefore suggested a number of heuristic methods for finding suitable partitions, and compared their performances in the context of Alg. 5 using randomly generated perturbed BMGs. We find that, even though good solutions for (B)PURC alone do not seem to be the most adequate approach, the value of the *UR*-costs appears most clearly in a combination with a method for community detection, more precisely, a modification of the Louvain method [29]. In particular, the results of our simulations suggest that realistic BMG data can be processed with sufficient accuracy and efficiency to make BMGs an attractive alternative to classical phylogenetic methods. The construction of bioinformatics workflows to process best hit data, e.g. at the first processing stage of `ProteinOrtho` [190], is a logical next step.

6

FROM BEST MATCHES TO ORTHOLOGY

The orthology relation has been characterized as a cograph by Hellmuth et al. [132] based on earlier work by Böcker and Dress [30]. This line of research has led to the idea of editing reciprocal best hit/match data to conform to the required cograph structure [133]. There are, however, two distinct sources of error in an orthology assignment pipeline based on best matches:

(i) inaccuracies in the assignment of best matches from sequence similarity data [293], and

(ii) limits in the reconstruction of the "true" orthology relation from best match graphs [103].

The estimation of best matches from similarity data has been discussed in Sec. 4.1.4 and [293], and we have investigated a further correction by means of arc modification in Chapter 5. In this chapter, we are only concerned with the second source of error. We, therefore, assume throughout that a "correct" BMG (cf. Def. 4.2) is given. *We do not assume, however, that we have any* a priori *knowledge about the underlying gene or species tree*. The problem we aim to solve is to determine the orthology relation that is best supported by the given BMG. Geiß et al. [103] showed that, for evolutionary scenarios that involve only speciations, gene duplications, and gene losses, there are no false-negative orthology assignments in the RBMG (see also Thm. 6.2 below). Our task, therefore, reduces to understanding the false-positive orthology assignments. *Importantly, we consider exclusively duplication-loss scenarios, i.e., we explicitly exclude horizontal gene transfer.*

This chapter is organized as follows: In Sec. 6.1, we formally introduce a reconciliation map for duplication-loss scenarios which was e.g. also used in [103] and present previous results that relate orthology and best matches in this framework. Sec. 6.2 formalizes the notion of *unambiguous false-positive* (*u-fp*) edges, i.e., reciprocal best matches that cannot be orthologs w.r.t. to *any* gene tree explaining the BMG. Secs. 6.3 and 6.4 contain the main mathematical contributions of this chapter:

1. We provide a full characterization of unambiguous false-positive orthology assignments in BMGs.
2. We provide a polynomial-time algorithm to determine all unambiguous false-positive orthology assignments in BMGs.

In Sec. 6.5, we complement the mathematical results with a computational analysis of simulated scenarios and observe that at least three-quarters of all false positives fall into this class. The remaining cases are not recognizable from best matches alone and correspond to complementary losses without surviving witnesses, i.e., cases that cannot be corrected without additional knowledge on the gene tree and/or the species tree. The results and ideas in this chapter appeared previously in [264].

Different Definitions and Notation in this Chapter

In biology, duplication events are not restricted to only occur after the first speciation event, i.e., the point in time when the last common ancestor of the species under consideration started diverging into two or more lineages. As alluded to in Sec. 2.3.6, we will assume that all trees in this chapter are planted to account for situations in which a (duplication) event occurs in the *ancestral line*. Recall that planted trees T are distinguished by a planted root 0_T whose unique child is the conventional root ρ_T.

In particular, it will be convenient to allow LRTs (T, σ) of BMGs to contain a planted edge $0_T\rho_T$ that is not contractible in the sense of least resolved trees (Def. 4.5) and redundant edges (Def. 4.6). Strictly speaking, this differs from the construction in [102, 104] and the previous chapters. However, there clearly is a one-to-one mapping between LRTs with and without an additional planted edge $0_T\rho_T$. Hence, the (non-contractible) planted edge is a trivial detail that does not affect the properties of LRTs. In particular, since $0_T\rho_T$ is not an inner edge by definition (cf. Sec. 2.3.6), the characterization of redundant edges in Lemma 4.6 does not need adjustment.

Throughout this and the following chapters, we will, whenever there is no danger of confusion, refer also to planted phylogenetic trees simply as trees.

In this chapter, we will furthermore consider RBMGs (G, σ) being the symmetric parts of BMGs $(\vec{G}, \sigma)$. In order to simplify the presentation, we will therefore interpret undirected graphs as the subclass of directed graphs G that satisfy $(x, y) \in E(G)$ if and only if $(y, x) \in E(G)$ for all $x, y \in V(G)$. Hence, a BMG $(\vec{G}, \sigma)$ is always a supergraph of its corresponding RBMG (G, σ), and thus, we can write $(G, \sigma) \subseteq (\vec{G}, \sigma)$. In turn, we will call bidirectional arcs $(x, y), (y, x) \in E(\vec{G})$ edges, and denote them by xy, even if $\vec{G}$ is not undirected according to this definition. Consequently, we say that an hourglass $[xy \mathrel{\nearrow\mkern-18mu\searrow} x'y']$ has edges xy and $x'y'$ (cf. Def. 4.10), and we will omit the arrow heads of all edges in the drawings in this chapter (see e.g. the BMG $(\vec{G}, \sigma)$ in Fig. 43).

Finally, we follow [103] and assume w.l.o.g. that the vertex- and leaf-colorings σ are surjective maps in this chapter to avoid species trees with superfluous leaves.

6.1 RECONCILIATION MAPS, EVENT-LABELING, AND ORTHOLOGY RELATIONS

Consider a gene tree (T, σ) and a corresponding species tree S, i.e., we have $\sigma(L(T)) = L(S)$. An *evolutionary scenario* extends the map $\sigma\colon L(T) \to L(S)$ to an embedding of the gene tree into the species tree. It (implicitly) describes different types of evolutionary events: speciations, gene duplications, and gene losses. As already mentioned, we do not consider other types of events

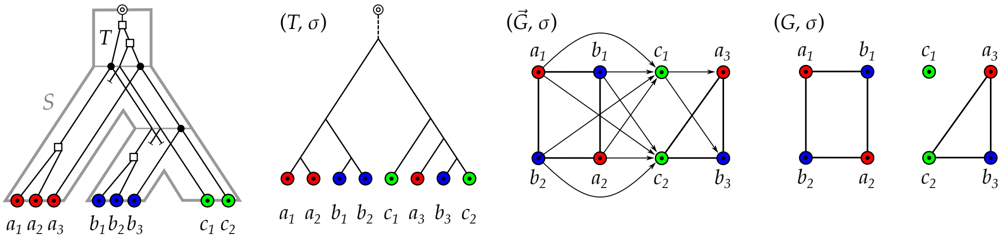


Fig. 43. An evolutionary scenario (left) with an observable part of the gene tree (T, σ) (second panel). See Fig. 6 and Def. 6.2 for an explanation of the symbols. The reciprocal best match graph (RBMG) (G, σ) on the right is the (undirected) symmetric part of the best match graph (BMG) $(\vec{G}, \sigma)$ (third panel).

such as horizontal gene transfer in this chapter. Gene losses do not appear explicitly since $L(T)$ only contains extant genes. Inner vertices in the gene tree T that designate speciations have their correspondence in inner vertices of the species tree. In contrast, gene duplications occur independently of speciations and thus belong to edges of the species tree. An example for an evolutionary scenario (with indicated loss events) and the corresponding BMG and RBMG is given in Fig. 43.

So far, a rather large number of axiom sets have been proposed and characterized to formalize reconciliations of gene and species trees, some of which are equivalent, see [82, 109, 130, 228, 252, 323] and the references therein. We will briefly discuss some of them in Sec. 7.1, especially such that include HGT. For duplication-loss scenarios and trees that are not necessarily binary, the embedding of T into S can be formalized by

Definition 6.1 (Reconciliation Map)**.** *Let S and T be two planted phylogenetic trees and let $\sigma\colon L(T) \to L(S)$ be a surjective map. A reconciliation from (T, σ) to S is a map $\mu\colon V(T) \to V(S) \cup E(S)$ satisfying*

(R0) Root Constraint. *$\mu(x) = 0_S$ if and only if $x = 0_T$.*

(R1) Leaf Constraint. *If $x \in L(T)$, then $\mu(x) = \sigma(x)$.*

(R2) Ancestor Preservation. *If $x \prec_T y$, then $\mu(x) \preceq_S \mu(y)$.*

(R3) Speciation Constraints. *Suppose $\mu(x) \in V^0(S)$ for some $x \in V(T)$. Then*

- *(i) $\mu(x) = \mathrm{lca}_S(\mu(v'), \mu(v''))$ for at least two distinct children v', v'' of x in T.*
- *(ii) $\mu(v')$ and $\mu(v'')$ are incomparable in S for any two distinct children v' and v'' of x in T.*

This type of reconciliation map has been established in [103]. Moreover, it has been shown in [103] that the axiom set used here is equivalent to axioms that are commonly used in the literature such as the DLS-trees of Górecki and Tiuryn [109] and reconciliation map introduced by Doyon et al. [82]. Axioms (R0) and (R1) are very simple constraints that determine the mapping of the planted root and the leaves of T, respectively. In particular, (R1) guarantees that μ is an extension of the leaf-coloring σ. The ancestor preservation constraint (R2) ensures that there are no conflicts between the ancestor relations $\preceq_T$ and $\preceq_S$ for any distinct $x, y \in V(T)$ and their images $\mu(x)$ and

$\mu(y)$. In [103], it was shown that the axioms imply the somewhat stronger condition $\mu(x) \prec_S \mu(y)$ if at least one of x and y is mapped to a vertex of S. Finally, the two conditions in (R3) are concerned with speciation vertices in T, i.e., vertices x that are mapped to inner vertices of the species tree. They ensure that any two children of x are mapped into incomparable branches below $\mu(x)$, and that, for at least two among them, the last common ancestor of their images in S is exactly the speciation $\mu(x)$.

Without any further constraints, Def. 6.1 gives rise to a well-known result [103, 115, 238]:

Lemma 6.1. [103, Lemma 3] *For every tree (T, σ) there is a reconciliation map μ to any species tree S with leaf set $L(S) = \sigma(L(T))$.*

The proof of Lemma 6.1 in [103] explicitly constructs a reconciliation map μ by simply applying Rules (R1) and (R2) and additionally mapping all inner vertices of T to the planted edge $0_S\rho_S$. Even though this usually implies a large number of loss events, and thus, seems unlikely to happen in biology, it is not ruled out logically by the axioms in Def. 6.1.

The reconciliation map μ from (T, σ) to S determines the types of evolutionary events in T. This can be formalized by associating an event labeling with the vertices of T. We use the notation introduced in [103]:

Definition 6.2. *Given a reconciliation map μ from (T, σ) to S, the* event labeling *on T* (determined by μ) *is the map $t_\mu : V(T) \to \{\circledcirc, \odot, \bullet, \square\}$ given by:*

$$t_\mu(u) = \begin{cases} \circledcirc & \text{if } u = 0_T, \text{ i.e., } \mu(u) = 0_S \text{ (root)} \\ \odot & \text{if } u \in L(T), \text{ i.e., } \mu(u) \in L(S) \text{ (leaf)} \\ \bullet & \text{if } \mu(u) \in V^0(S) \text{ (speciation)} \\ \square & \text{else, i.e., } \mu(u) \in E(S) \text{ (duplication)} \end{cases}$$

While every gene tree can be reconciled with any species tree, this is no longer true if event labels are prescribed in the gene tree T [130, 140, 180]. The following result is a simple but useful consequence of combining the axioms of the reconciliation map with the event labeling of Def. 6.2.

Lemma 6.2. [103, Lemma 3] *Let μ be a reconciliation map from (T, σ) to a tree S and suppose that $u \in V(T)$ is a vertex with $\mu(u) \in V^0(S)$ and thus, $t(\mu(u)) = \bullet$. Then, $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) = \emptyset$ for any two distinct $v_1, v_2 \in \mathsf{child}_T(u)$.*

We will regularly make use of the observation that, by contraposition of Lemma 6.2, $\sigma(L(T(v))) \cap \sigma(L(T(v'))) \neq \emptyset$ for two distinct $v_1, v_2 \in \mathsf{child}_T(u)$ implies that $\mu(u) \in E(S)$, and thus $t_\mu(u) = \square$. Lemma 6.2 suggests to define *event-labeled trees* as trees (T, t) endowed with a map $t : V(T) \to \{\circledcirc, \odot, \bullet, \square\}$ such that $t(0_T) = \circledcirc$ and $t(u) = \odot$ for all $u \in L(T)$. In [103], Lemma 6.2 also served as a motivation for

Definition 6.3. *Let* (T,σ) *be a leaf-colored tree. The* extremal event labeling *of* T *is the map* $\widehat{t}_T : V(T) \to \{\circledcirc, \odot, \bullet, \square\}$ *defined for* $u \in V(T)$ *by*

$$\widehat{t}_T(u) = \begin{cases} \circledcirc & \text{if } u = 0_T \\ \odot & \text{if } u \in L(T) \\ \square & \text{if there are two children } v_1, v_2 \in \mathsf{child}_T(u) \text{ such that} \\ & \qquad \sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) \neq \emptyset \\ \bullet & \text{otherwise} \end{cases}$$

The extremal event labeling $\widehat{t}_T$ of (T,σ) is the one that minimizes the necessary number of duplications on (T,σ) in the light of Lemma 6.2. In a conceptual sense, therefore, $(T,\widehat{t}_T)$ is a "most parsimonious" solution, matching the idea of most parsimonious reconciliations [115, 238]. An example of an extremal event labeling is shown in Fig. 53 on page 180 (rightmost tree). The extremal event labeling is closely related to the concept of apparent duplication (AD) vertices often found in the literature [e.g. 183, 304]. For a (binary) gene tree T and a reconciliation of T with a species tree S, a duplication vertex of T is an AD vertex if its two subtrees have at least one color in common. In contrast, it is a non-apparent duplication (NAD) vertex if the color sets of its subtrees are disjoint. This notion is useful for a variety of parsimony problems that usually aim to avoid or minimize the number of NAD vertices [183, 304]. However, the extremal event labeling $\widehat{t}_T$ is completely defined by (T,σ). That is, in contrast to both the event labeling in Def. 6.2 and the concept of AD and NAD vertices, $\widehat{t}_T$ does not depend on a specific reconciliation map. On the other hand, there is no guarantee that there always exists a reconciliation map μ from (T,σ) to some species tree S such that $t_\mu = \widehat{t}_T$, cf. [103, Fig. 2] and Fig. 53 in Sec. 6.3.7 for counterexamples. Nevertheless, we shall see below that the extremal labeling is a key step towards identifying false-positive orthology assignments.

The event labeling on T defines the orthology graph.

Definition 6.4. *The* orthology graph $\Theta(T,t)$ *of an event-labeled tree* (T,t) *has vertex set* $L(T)$ *and edges* $uv \in E(\Theta)$ *if and only if* $t(\mathsf{lca}_T(u,v)) = \bullet$.

The orthology graph is often referred to as the orthology relation. Hellmuth et al. [132] showed that orthology graphs coincide with cographs:

Theorem 6.1. [132, Cor. 4] *A graph* G *is an orthology graph for some event-labeled tree* (T,t)*, i.e.* $G = \Theta(T,t)$*, if and only if* G *is a cograph.*

In particular, therefore, orthology graphs do not contain induced paths on four vertices (P_4s), cf. Prop. 2.6.

The orthology graph is a subgraph of the RBMG (and thus also of the BMG) for any given reconciliation map connecting a gene with a species tree.

Theorem 6.2. [103, Lemmas 4 and 5] *Let* (T,σ) *be a leaf-colored tree and* μ *a reconciliation map from* (T,σ) *to some species tree* S*. Then* $\Theta(T,t_\mu) \subseteq \Theta(T,\widehat{t}_T) \subseteq G(T,\sigma) \subseteq \vec{G}(T,\sigma)$.

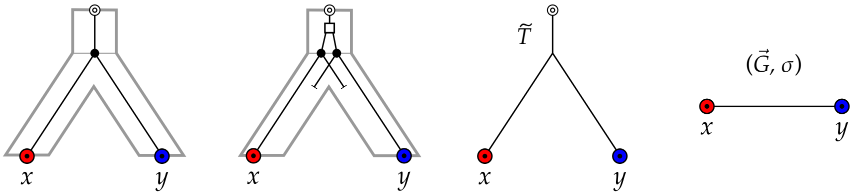


Fig. 44. Two scenarios (1st and 2nd panel to the left) for the evolution of a gene family embedded into a species tree (shown in gray), where ● represents speciation and □ duplication events. The second scenario is the simplest example for a complementary gene loss that is not witnessed by any other species. In particular, the two different true histories result in the same topology $\widetilde{T}$ of the true (loss-free) gene tree, and thus explain the same BMG $(\vec{G},\sigma)$. However, only for the leftmost scenario the edge xy in $(\vec{G},\sigma)$ describes correct orthologs.

In particular, $t_\mu(v) = \bullet$ implies $\widehat{t}_T(v) = \bullet$ for any reconciliation map. By contraposition, therefore, if $\widehat{t}_T(v) = \square$, then $t_\mu(v) = \square$ for all possible reconciliation maps μ from (T,σ) to any species tree S. A crucial implication of Thm. 6.2 is that edges in a BMG $\vec{G}(T,\sigma)$ always correspond to either correct orthologous pairs of genes or false-positive orthology assignments. Hence, $\vec{G}(T,\sigma)$ never contains false-negative orthology assignments.

6.2 FALSE-POSITIVE ORTHOLOGY ASSIGNMENTS

Of course, the *true* orthology relation is not known. Nevertheless, we start our mathematical analysis with the following definition: Assume that $(\vec{G},\sigma)$ is the BMG (with corresponding RBMG (G,σ)) deriving from a duplication-loss scenario that is unknown to us. Denote by $(\widetilde{T},\widetilde{t},\sigma)$ the corresponding true leaf-colored and event-labeled gene tree. A pair of genes x and y that are not true orthologs ($xy \notin E(\Theta(\widetilde{T},\widetilde{t}))$) but reciprocal best matches ($xy \in E(G)$) are false-positive orthologs. If they are orthologs ($xy \in E(\Theta(\widetilde{T},\widetilde{t}))$) but not reciprocal best matches ($xy \notin E(G)$), they are false-negative orthologs. By Thm. 6.2, (G,σ) cannot contain false-negative orthology assignments. Being a false positive is a property of the edge xy in an RBMG, and equivalently of the symmetric pair (x,y) and (y,x) in the BMG. Here, we aim to identify false-positive edges from the structure of the BMG itself.

We first note that false positives cannot be avoided altogether, i.e., not all false positives can be identified from a BMG alone. The simplest example, Fig. 44 (second scenario), comprises a gene duplication and a subsequent speciation and complementary gene losses in the descendant lineages such that each paralog survives only in one of them. In this situation, xy is a reciprocal best match. If there are no other descendants that harbor genes witnessing the duplication event, then the framework of best matches provides no information to recognize xy as a false-positive assignment.

On the other hand, RBMGs and thus BMGs contain at least some information on false positives. Since the orthology relation forms a cograph but RBMGs are not cographs in general [104], incorrect orthology assignments are associated with induced P_4s, the forbidden subgraphs that characterize cographs (cf. Prop. 2.6). P_4s arise for instance as a consequence of the complete loss of different paralogous groups in disjoint lineages. Dessimoz et al. [75] noted that such false-positive orthology assignments can be identified under certain circumstances, in particular, if there is some species in which both paralogs have survived. The corresponding motif in BMGs, the "good quartets", was investigated in some detail by Geiß et al. [104]. The removal of such false-positive orthologs already leads to a substantial improvement of the orthology assignments in simulated data [103]. Here, we extend the results of Geiß et al. [103] to a complete characterization of false-positive orthology assignments for a given BMG. Importantly, good quartets cannot be defined on RBMGs alone because information on non-reciprocal best matches is also needed explicitly.

As discussed above, we are not concerned here with the errors that arise in the reconstruction of best matches from sequence similarity data. We therefore assume that we are given a BMG $(\vec{G}, \sigma)$ as specified in Def. 4.2. We assume no additional information about the gene tree or the species tree, i.e., the only data about the evolutionary scenario that is available to us is the BMG $(\vec{G}, \sigma)$.

In order to study false-positive orthology assignments, we first consider a tree (T, σ) that explains the BMG $(\vec{G}, \sigma)$. We neither make the assumption that (T, σ) is least resolved nor that (T, σ) reflects the true history, i.e., that (T, σ) is related to the true gene tree $(\widetilde{T}, \sigma)$.

Definition 6.5 ((T,σ)-false-positive)**.** *Let (T, σ) be a tree explaining the BMG $(\vec{G}, \sigma)$. An edge xy in $\vec{G}$ is called* (T,σ)-false-positive*, or* (T,σ)-fp *for short, if for every reconciliation map μ from (T, σ) to any species tree S we have $t_\mu(\mathrm{lca}_T(x, y)) = \square$, i.e., $\mu(\mathrm{lca}_T(x, y)) \in E(S)$.*

In other words, xy is called *(T,σ)-fp* whenever x and y cannot be orthologous w.r.t. any possible reconciliation μ from (T, σ) to any species tree. Interestingly, *(T,σ)-fp*s can be identified without considering reconciliation maps explicitly.

Lemma 6.3. *Let $(\vec{G}, \sigma)$ be a BMG, xy be an edge in $\vec{G}$ and (T, σ) be a tree that explains $(\vec{G}, \sigma)$. Then, the following statements are equivalent:*

1. *The edge xy is (T,σ)-fp.*
2. *There are two children v_1 and v_2 of $\mathrm{lca}_T(x, y)$ such that $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) \neq \emptyset$.*
3. *For the extremal labeling $\widehat{t}_T$ of (T, σ) it holds that $\widehat{t}_T(\mathrm{lca}_T(x, y)) = \square$.*

Proof. (2) implies (1). Suppose that there are two children v_1 and v_2 of $\mathrm{lca}_T(x, y)$ such that $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) \neq \emptyset$. By Lemma 6.2, $\mu(\mathrm{lca}_T(x, y)) \in E(S)$ and thus, $t_\mu(\mathrm{lca}_T(x, y)) = \square$ for all possible reconciliation maps μ from (T, σ) to any species tree S. Hence, xy is *(T,σ)-fp*.

(1) implies (2). By contraposition, let $v = \mathrm{lca}_T(x, y)$ and suppose that for all distinct children $v_i, v_j \in \mathsf{child}_T(v) = \{v_1, \dots, v_k\}$, $k \geq 2$ we have $\sigma(L(T(v_i))) \cap \sigma(L(T(v_j))) = \emptyset$. In the following, we show that there is a species tree S and a reconciliation map μ from (T, σ) to S such that $t_\mu(\mathrm{lca}_T(x, y)) = \bullet$, which implies that xy is not *(T,σ)-fp*.

We construct the species tree S as follows: S has root edge $0_S\rho_S$. Now add k children $u_1, \dots, u_k$ to ρ_S. For each of these children u_i with $|\sigma(L(T(v_i)))| > 1$, we add a leaf t for every color $t \in \sigma(L(T(v_i)))$ and the edge $u_i t$. Any other u_i is considered to be a leaf in S, and we identify u_i with the single element in $\sigma(L(T(v_i)))$. Furthermore, add for all $t \in \sigma(L(T)) \setminus \sigma(L(T(v)))$ a leaf t that is adjacent to ρ_S. Since the color sets $\sigma(L(T)) \setminus \sigma(L(T(v))), \sigma(L(T(v_1))), \dots, \sigma(L(T(v_k))$ are pairwise distinct, S is well-defined, and, by construction, a planted phylogenetic tree. To construct a reconciliation map we put (i) $\mu(0_T) = 0_S$; (ii) $\mu(x) = \sigma(x)$ for all $x \in L(T)$; (iii) $\mu(v) = \rho_S$; (iv) $\mu(w) = 0_S\rho_S$ for all $w \in V^0(T \setminus T(v))$; and (v) $\mu(w) = \rho_S u_i$ for all $w \in V^0(T(v_i))$. By Condition (i) and (ii), the Axioms (R0) and (R1) are satisfied, respectively. By Condition (v), we have $\mu(v_i) = \rho_S u_i$ if v_i is an inner vertex. Otherwise, v_i is a leaf and $|\sigma(L(T(v_i)))| = 1$. Therefore, $\mu(v_i) = \sigma(v_i) = u_i$ by (ii) and by construction. It is easy to verify that μ satisfies (R2). A sketch of construction of the species tree S and the reconciliation map μ is provided in Fig. 45.

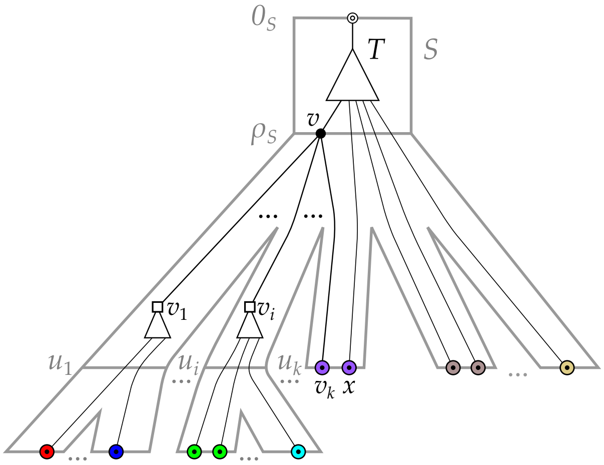


Fig. 45. Visualization of the construction of a species tree S and reconciliation map μ as described in the proof of Lemma 6.3. Note that, in the example, v_k is already a leaf in the gene tree T. Hence, the corresponding u_k is also a leaf since $|\sigma(L(T(v_k)))| = 1$. Moreover, note that for $x \in L(T) \setminus L(T(v))$, it is possible that $\mu(x) = u_j$ or $\mu(x) = t$ with $t \in \mathsf{child}_S(u_j)$ for some u_j.

The only vertex of T that is mapped to a vertex in S is v. Hence, it remains to show that $\mu(v) = \rho_S \in V^0(S)$ satisfies (R3). Note that for every two distinct children v_i, v_j of v we have $\mu(v_i) \in \{\rho_S u_i, u_i\}$ and $\mu(v_j) \in \{\rho_S u_j, u_j\}$. In any case, $\mu(v_i)$ and $\mu(v_j)$ are incomparable in S. Hence, (R3.ii) is satisfied. In particular, $\mu(v) = \rho_S = \mathrm{lca}_S(\mu(v_i), \mu(v_j))$ for all distinct $v_i, v_j \in \mathsf{child}_T(v)$. Hence, (R3.i) is satisfied. In summary, μ is a reconciliation map from (T, σ) to S. Since $\mu(v) = \rho_S \in V^0(S)$, we have $t_\mu(v) = \bullet$.

Statements (2) and (3) are equivalent by definition of the extremal event labeling. ☐

Lemma 6.3 implies that (T,σ)-*fp* can be verified in polynomial time for any given gene tree (T, σ). By contraposition of Lemma 6.2, inner vertices with two distinct children v_1 and v_2 satisfying $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) \neq \emptyset$ are duplication vertices for every possible reconciliation map to every possible species tree. Therefore, the property of being an AD vertex only depends on (T, σ). In particular, (T,σ)-*fp* edges coincide with the edges xy in $(\vec{G}, \sigma)$ for which $\mathrm{lca}_T(x, y)$ is an AD vertex.

As shown in Fig. 46, there are trees (T_1, σ) and (T_2, σ) that explain the same BMG for which, however, the edges xz, $x'z$, and yz are (T_1,σ)-*fp* but not

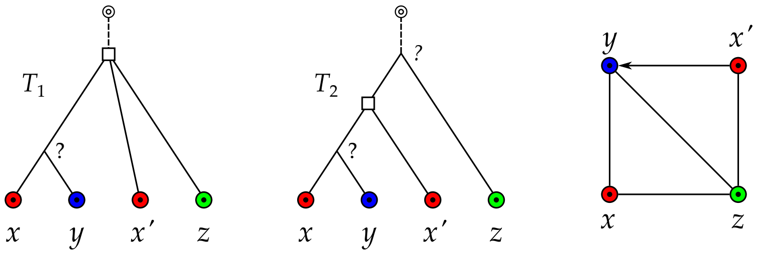


Fig. 46. The BMG $(\vec{G},\sigma)$ shown on the right is explained by both (T_1,σ), which is the unique least resolved tree for $(\vec{G},\sigma)$, and (T_2,σ). The vertices labeled $\square$ must be duplications due to Lemma 6.2, whereas the vertices labeled "?" could be both duplications or speciations. The edges xz, $x'z$ and yz are (T_1,σ)-*fp* but not (T_2,σ)-*fp* (cf. Lemma 6.3). Thus, neither of the edges xz, $x'z$ and yz is *u-fp*.

(T_2,σ)-*fp*. Since we assume that no information on (T,σ) is available *a priori*, it is natural to consider the set of edges that are false positives for all trees explaining a given BMG.

Definition 6.6 (Unambiguous false-positive). *Let $(\vec{G},\sigma)$ be a BMG. An edge xy in $\vec{G}$ is called* unambiguous false-positive (*u-fp*) *if for all trees (T,σ) that explain $(\vec{G},\sigma)$ the edge xy is (T,σ)*-fp.

Hence, if an edge xy in $(\vec{G},\sigma)$ is *u-fp*, then it is in particular (T,σ)-*fp* in the true history that explains $(\vec{G},\sigma)$. Thus, *u-fp* edges are always correctly identified as false positives. Not all "correct" false-positive edges are *u-fp*, however. It is possible that, for an edge xy in $\vec{G}$, we have $t_\mu(\mathrm{lca}_T(x,y)) = \square$ for the true gene tree and the true species tree, but xy is not (T',σ)-*fp* for some gene tree (T',σ) possibly different from (T,σ). One of the simplest examples is shown in Fig. 44, assuming that $(\vec{G},\sigma)$ is the "true" BMG. Since $t_\mu(\mathrm{lca}_{\widetilde{T}}(x,y)) = \bullet$ may be possible (Fig. 44, leftmost scenario) the edge xy is not $(\widetilde{T},\sigma)$-*fp* and therefore not *u-fp*.

6.3 CHARACTERIZATION OF UNAMBIGUOUS FALSE-POSITIVE (*u-fp*) EDGES

In order to adapt the concept of AD vertices for our purposes, we introduce the color-intersection $\mathcal{S}^{\cap}$ associated with a gene tree (T,σ). For a pair of distinct leaves $x,y \in L(T)$ we denote by $v_x, v_y \in \mathrm{child}_T(\mathrm{lca}_T(x,y))$ the unique children of the last common ancestor of x and y for which $x \preceq_T v_x$ and $y \preceq_T v_y$. That is, $T(v_x)$ and $T(v_y)$ are the subtrees of T rooted in the children of $\mathrm{lca}_T(x,y)$ with $x \in L(T(v_x))$ and $y \in L(T(v_y))$. The set

$$\mathcal{S}^{\cap}_T(x,y) := \sigma(L(T(v_x))) \cap \sigma(L(T(v_y)))$$

contains the colors, i.e. species, that are common to both subtrees. In particular, the existence of common colors, $\mathcal{S}^{\cap}_T(x,y) \neq \emptyset$, determines whether or not the inner vertex $\mathrm{lca}_T(x,y)$ is AD. Lemma 4.4 immediately implies

Corollary 6.1. *Let xy be an edge in a BMG $(\vec{G},\sigma)$. Then $\sigma(\{x,y\}) \cap \mathcal{S}^{\cap}_T(x,y) = \emptyset$ for all trees (T,σ) that explain $(\vec{G},\sigma)$.*

The following result shows that the color-intersection of a given edge in a BMG $(\vec{G},\sigma)$ in fact does not depend on the tree representation of $(\vec{G},\sigma)$.

Lemma 6.4. *Let $(\vec{G},\sigma)$ be a BMG and (T^*,σ) the corresponding unique least resolved tree explaining $(\vec{G},\sigma)$. Then, for each tree (T,σ) that explains $(\vec{G},\sigma)$, every edge xy in $(\vec{G},\sigma)$ satisfies $\mathcal{S}^{\cap}_{T^*}(x,y) = \mathcal{S}^{\cap}_{T}(x,y)$. Thus, in particular, $\mathcal{S}^{\cap}_{T^*}(x,y) \neq \emptyset$ if and only if $\mathcal{S}^{\cap}_{T}(x,y) \neq \emptyset$.*

Proof. Let (T,σ) be an arbitrary tree that explains $(\vec{G},\sigma)$. Moreover, let xy be an edge in $\vec{G}$ and denote by v_x and v_y be the unique children $v_x, v_y \in \mathsf{child}_T(\mathrm{lca}_T(x,y))$ with $x \preceq_T v_x$ and $y \preceq_T v_y$. Analogously, v^*_x and v^*_y are the unique children $v^*_x, v^*_y \in \mathsf{child}_{T^*}(\mathrm{lca}_{T^*}(x,y))$ with $x \preceq_{T^*} v^*_x$ and $y \preceq_{T^*} v^*_y$.

First, we show that $t \in \mathcal{S}^{\cap}_{T^*}(x,y)$ implies $t \in \mathcal{S}^{\cap}_{T}(x,y)$. Since (T,σ) explains $(\vec{G},\sigma)$, we apply Thm. 4.1 to conclude that T is a refinement of T^* and thus, $\mathcal{C}(T^*) \subseteq \mathcal{C}(T)$. Therefore, $L(T^*(\mathrm{lca}_{T^*}(x,y)))$, $L(T^*(v^*_x))$ and $L(T^*(v^*_y))$ are contained in $\mathcal{C}(T)$. This implies that there must be vertices u, w_x, and w_y in T with $L(T(u)) = L(T^*(\mathrm{lca}_{T^*}(x,y)))$, $L(T(w_x)) = L(T^*(v^*_x))$ and $L(T(w_y)) = L(T^*(v^*_y))$. Note that $L(T^*(v^*_x)) \cap L(T^*(v^*_y)) = \emptyset$, and thus $L(T(w_x)) \cap L(T(w_y)) = \emptyset$. In particular, w_x and w_y are incomparable in T. Moreover, $u = \mathrm{lca}_T(x,y) = \mathrm{lca}_T(w_x,w_y)$, thus we have $w_x \preceq_T v_x$ and $w_y \preceq_T v_y$. Therefore, $L(T^*(v^*_x)) \subseteq L(T(v_x))$ and $L(T^*(v^*_y)) \subseteq L(T(v_y))$. Therefore, $t \in \mathcal{S}^{\cap}_{T^*}(x,y)$ implies $t \in \mathcal{S}^{\cap}_{T}(x,y)$.

Now, we show that $t \in \mathcal{S}^{\cap}_{T}(x,y)$ implies $t \in \mathcal{S}^{\cap}_{T^*}(x,y)$. Let $t \in \mathcal{S}^{\cap}_{T}(x,y) \neq \emptyset$. In this case, $t \in \sigma(L(T(v_x)))$ and we can choose a vertex $z_1 \in L(T(v_x))$ such that $\sigma(z_1) = t$ and $\mathrm{lca}_T(x,z_1)$ is as far away as possible from v_x compared to all $\mathrm{lca}_T(x,z)$ with $z \in L[t]$, i.e., $\mathrm{lca}_T(x,z_1) \preceq_T \mathrm{lca}_T(x,z)$ for all $z \in L[t]$. Thus, $(x,z_1) \in E(\vec{G})$. An analogous argument ensures that there is a vertex $z_2 \in L(T(v_y))$ such that $\sigma(z_2) = t$ and $(y,z_2) \in E(\vec{G})$. Clearly, $\mathrm{lca}_T(x,z_2) = \mathrm{lca}_T(x,y) = \mathrm{lca}_T(y,z_1)$ and thus $\mathrm{lca}_T(x,z_1) \preceq_T v_x \prec_T \mathrm{lca}_T(x,z_2)$, which in turn implies that $(x,z_2) \notin E(\vec{G})$. Since $(x,z_1) \in E(\vec{G})$ and $(x,z_2) \notin E(\vec{G})$, we obtain the informative triple $xz_1|z_2$ for $(\vec{G},\sigma)$. Analogously, $yz_2|z_1$ is an informative triple for $(\vec{G},\sigma)$. Lemma 4.5 and the fact that T^* explains $(\vec{G},\sigma)$ implies that there are distinct vertices $v_1, v_2 \in \mathsf{child}_{T^*}(\mathrm{lca}_{T^*}(x,y))$ such that $x, z_1 \preceq_{T^*} v_1$ and $y, z_2 \preceq_{T^*} v_2$. Since $t = \sigma(z_1) = \sigma(z_2)$, we have $t \in \mathcal{S}^{\cap}_{T^*}(x,y)$.

Finally, $t \in \mathcal{S}^{\cap}_{T^*}(x,y)$ if and only if $t \in \mathcal{S}^{\cap}_{T}(x,y)$ implies both $\mathcal{S}^{\cap}_{T^*}(x,y) = \mathcal{S}^{\cap}_{T}(x,y)$ and $\mathcal{S}^{\cap}_{T^*}(x,y) \neq \emptyset$ *if and only if* $\mathcal{S}^{\cap}_{T}(x,y) \neq \emptyset$. ☐

Remark 6.1. *By Lemma 6.4, we have $\mathcal{S}^{\cap}_{T}(x,y) = \mathcal{S}^{\cap}_{T^*}(x,y)$ for every tree (T,σ) explaining a BMG $(\vec{G},\sigma)$ with corresponding least resolved tree (T^*,σ). Therefore, it is sufficient to consider $\mathcal{S}^{\cap}_{T^*}(x,y)$. We will therefore drop the explicit reference to the tree and simply write $\mathcal{S}^{\cap}(x,y)$. We can verify in polynomial time whether or not $\mathcal{S}^{\cap}(x,y) = \emptyset$ because the least resolved tree (T^*,σ) explaining $(\vec{G},\sigma)$ can be computed in polynomial time (cf. Cor. 4.6).*

Proposition 6.1. *Every edge xy in a BMG $(\vec{G},\sigma)$ with $\mathcal{S}^{\cap}(x,y) \neq \emptyset$ is* u-fp.

Proof. By Lemma 6.4 and Remark 6.1, $\mathcal{S}^{\cap}(x,y) \neq \emptyset$ if and only if $\mathcal{S}^{\cap}_{T}(x,y) \neq \emptyset$ for all trees (T,σ) that explain $(\vec{G},\sigma)$. By Lemma 6.2, $\mu(\mathrm{lca}_T(x,y)) \in E(S)$ and thus, $t_\mu(\mathrm{lca}_T(x,y)) = \square$ for all trees (T,σ) that explain $(\vec{G},\sigma)$. Hence, xy is *u-fp*. ☐

An immediate consequence of Prop. 6.1 is:

Corollary 6.2. *An edge xy in a BMG $\vec{G}(T,\sigma)$ with $\mathcal{S}^{\cap}(x,y) \neq \emptyset$ is* (T,σ)-fp.

As we shall see later, the converse of Prop. 6.1 and Cor. 6.2 is not always satisfied (cf. also Fig. 48). However, we show next that it does hold for the special case of binary trees.

Lemma 6.5. *Let xy be an edge in $\vec{G}(T,\sigma)$ and suppose $\operatorname{lca}_T(x,y)$ is a binary vertex. Then, the following three statements are equivalent:*

1. *The edge xy is (T,σ)-*fp.
2. *$\mathcal{S}^{\cap}(x,y) \neq \emptyset$.*
3. *The edge xy is* u-fp.

Proof. (1) implies (2). Suppose xy is (T,σ)*-fp*. Since v is binary, it has precisely two children v_1 and v_2. In particular, $v = \operatorname{lca}_T(x,y)$ implies that that $x \preceq_T v_i$ and $x \preceq_T v_j$ for $i,j \in \{1,2\}$ being distinct. By Lemma 6.3, the two children v_1 and v_2 of v satisfy $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) \neq \emptyset$. By Lemma 6.4 and Remark 6.4, we have $\mathcal{S}^{\cap}(x,y) \neq \emptyset$.
(2) implies (3). If $\mathcal{S}^{\cap}(x,y) \neq \emptyset$, we can apply Prop. 6.1 to conclude that xy is *u-fp*.
(3) implies (1). By definition, if xy is *u-fp*, then it is in particular also (T,σ)*-fp*. □

Theorem 6.3. *Let $(\vec{G},\sigma)$ be a BMG that is explained by a binary tree (T,σ). Then, for every edge xy in $(\vec{G},\sigma)$, the following three statements are equivalent:*

1. *The edge xy is (T,σ)-*fp.
2. *$\mathcal{S}^{\cap}(x,y) \neq \emptyset$.*
3. *The edge xy is* u-fp.

Proof. For every edge xy in $\vec{G}$ the last common ancestor $\operatorname{lca}_T(x,y)$ is binary. Now apply Lemma 6.5. □

Thm. 6.3 implies that all *u-fp* edges can be detected in a BMG that is explained by a known binary gene tree. If existent, such a tree can be constructed in polynomial time as a consequence of Cor. 4.10. However, Thm. 6.3 does not generalize to the non-binary case, and $\mathcal{S}^{\cap}(x,y)$ is not sufficient to identify all *u-fp* edges. Furthermore, it is not difficult to find non-binary trees in which (T,σ)*-fp* and *u-fp* edges are not the same: As show in Fig. 46, the edge xz in is (T_1,σ)*-fp* but not (T_2,σ)*-fp* according to Lemma 6.3. Since both trees explain the same BMG, the edge xy is not *u-fp*.

Recall that Prop. 4.8 in Sec. 4.3 provides a characterization of BMGs that can be explained by binary trees as such BMGs that are hourglass-free; a property that can be tested in polynomial time (cf. Cor. 4.9). We shall see in Sec. 6.3.2 that hourglasses play a central role in the identification of additional *u-fp* edges.

6.3.1 u-fp *Edges in Quartets – The Case $\mathcal{S}^{\cap}(x,y) \neq \emptyset$*

Since every orthology graph is a cograph (Thm. 6.1) and thus free of induced P_4s, every induced P_4 in the RBMG necessarily contains a false-positive orthology assignments. The subgraphs of the BMG spanned by a P_4 in its symmetric part (i.e., the RBMG) are known as quartets. The quartets on three colors of a BMG $(\vec{G},\sigma)$ fall into three distinct classes depending on the coloring and the additional, non-symmetric edges (cf. [104, Lemma 32]). We write $\langle abcd \rangle$ or, equivalently, $\langle dcba \rangle$ for an induced P_4 with edges ab, bc, and cd. The quartets on three colors fall into three classes:

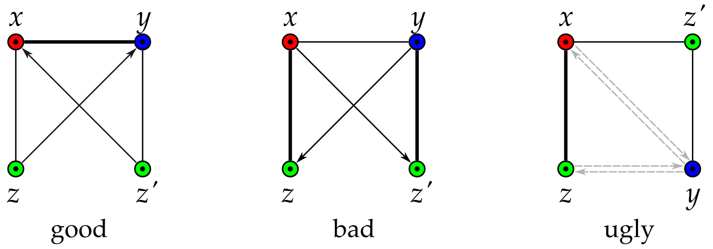


Fig. 47. The three types of quartets in BMGs. Ugly quartets may or may not contain either of the two (dashed) arcs between x and y, and y and z, respectively. Bold edges highlight the middle and first edges of the respective quartets as specified in Def. 6.7.

Definition 6.7 (Good, bad, and ugly quartets). *Let $(\vec{G},\sigma)$ be a BMG with symmetric part (G,σ) and vertex set L, and let $Q := \{x,y,z,z'\} \subseteq L$ with $x \in L[r]$, $y \in L[s]$, and $z,z' \in L[t]$. The set Q, resp., the induced subgraph $(\vec{G}[Q],\sigma_{|Q})$ is*

- *a* good quartet *if (i) $\langle zxyz'\rangle$ is an induced P_4 in (G,σ) and (ii) $(z,y),(z',x) \in E(\vec{G})$ and $(y,z),(x,z') \notin E(\vec{G})$,*
- *a* bad quartet *if (i) $\langle zxyz'\rangle$ is an induced P_4 in (G,σ) and (ii) $(y,z),(x,z') \in E(\vec{G})$ and $(z,y),(z',x) \notin E(\vec{G})$,*
- *an* ugly quartet *if $\langle zxz'y\rangle$ is an induced P_4 in (G,σ).*

The edge xy in a good quartet $\langle zxyz'\rangle$ is its middle *edge. The edge zx of an ugly quartet $\langle zxz'y\rangle$ or a bad quartet $\langle zxyz'\rangle$ is called its* first *edge. First edges in ugly quartets are uniquely determined due to the colors. In bad quartets, this is not the case and therefore, the edge yz' in $\langle zxyz'\rangle$ is a first edge as well.*

The three different types of quartets are shown in Fig. 47. RBMGs never contain induced P_4s on two colors [104, Obs. 5]. This, in particular, implies that for the induced P_4s in Def. 6.7 the colors r, s, and t must be pairwise distinct. Note that (R)BMGs may also contain induced P_4s on four colors. These are investigated in some more detail in Sec. 6.4.3.

Good quartets are characteristic of a complementary gene loss (as shown in Fig. 44) that is "witnessed" by a third species in which both child branches of the problematic duplication event survive. That is, good quartets appear if there is a pair of genes z and z' with $\sigma(z) = \sigma(z')$ and $\mathrm{lca}_T(z,z') = \mathrm{lca}_T(x,y)$ in the true gene tree. We remark that previous work also noted that complementary gene loss can be resolved successfully under certain circumstances [75] such as this one. An in-depth analysis of quartets shows that they can be used to identify many of the *u-fp* edges.

The key property of good quartets is a consequence of [103, Cor. 5], and restated here as:

Proposition 6.2. *If $\langle zxyz'\rangle$ is a good quartet in the BMG $(\vec{G},\sigma)$, then $\mathcal{S}^{\cap}(x,y) \neq \emptyset$ and thus, xy is* u-fp.

Proof. Let $\langle zxyz'\rangle$ in $(\vec{G},\sigma)$ be a good quartet in $(\vec{G},\sigma)$ and let (T,σ) be an arbitrary tree explaining $(\vec{G},\sigma)$. Then [104, Lemma 36] implies that $v := \mathrm{lca}_T(x,y,z,z')$ has two distinct children $v_1, v_2 \in \mathsf{child}_T(v)$ such that $x,z \preceq_T v_1$ and $y,z' \preceq_T v_2$. Hence,

$v = \text{lca}_T(x,y)$. Since $\sigma(z) \in \sigma(L(T(v_1))) \cap \sigma(L(T(v_2)))$, we have $\mathcal{S}^{\cap}(x,y) \neq \emptyset$ and, by Prop. 6.1, the edge xy is *u-fp*. $\square$

Prop. 6.2 provides a convenient way to identify unambiguous false-positive edges in a BMG.

Lemma 6.6. *If xy is an edge in a BMG $\vec{G}(T,\sigma)$ and $t \in \mathcal{S}^{\cap}(x,y)$, then there is a good quartet $\langle z_1x^*y^*z_2\rangle$ such that*

(a) $\sigma(x^*) = \sigma(x)$, $\sigma(y^*) = \sigma(y)$, and $\sigma(z_1) = \sigma(z_2) = t$;

(b) $x^*, z_1 \in L(T(v_x))$ and $y^*, z_2 \in L(T(v_y))$ with v_x and v_y being the unique children in $\text{child}_T(\text{lca}_T(x,y))$ such that with $x \preceq_T v_x$ and $y \preceq_T v_y$.

Proof. Consider an edge xy of $\vec{G}(T,\sigma)$ and a color $t \in \mathcal{S}^{\cap}(x,y)$. By Cor. 6.1, $t \neq \sigma(x), \sigma(y)$. Lemma 4.3 ensures the existence of an edge x^*z_1 in $\vec{G}$ for some leaves $x^* \in L(T(v_x)) \cap L[\sigma(x)]$ and $z_1 \in L(T(v_x)) \cap L[t]$. By the same arguments as in the proof of Cor. 6.1, we can conclude that z_1y' is not an edge in $\vec{G}$ for all $y' \in L(T(v_y)) \cap L[\sigma(y)]$. However, $(z_1, y') \in E(\vec{G})$ since the color of y' is not present in $T(v_x)$. Likewise, there are leaves $y^* \in L(T(v_y)) \cap L[\sigma(y)]$ and $z_2 \in L(T(v_y)) \cap L[t]$ such that y^*z_2 forms an edge in $\vec{G}$. Reusing the arguments from $L(T(v_x))$, we find that $x'z_2$ is not an edge in $\vec{G}$ and $(z_2, x') \in E(\vec{G})$ for any $x' \in L(T(v_x)) \cap L[\sigma(x)]$. Finally, $\sigma(x) \notin \sigma(L(T(v_y)))$ and $\sigma(y) \notin \sigma(L(T(v_x)))$ implies that x^*y^* forms an edge in $\vec{G}$. Hence, $\langle z_1x^*y^*z_2\rangle$ is a good quartet. $\square$

The edge x^*y^* in Lemma 6.6 is the middle edge of a good quartet. For completeness, we also provide a result for the identification of *u-fp* edges using bad quartets:

Proposition 6.3. *Let $\langle zxyz'\rangle$ be a bad quartet in a BMG $(\vec{G},\sigma)$. Then, the edges xz and yz' are* u-fp *and every tree that explains $(\vec{G},\sigma)$ is non-binary.*

Proof. Let (T,σ) be an arbitrary tree that explains $(\vec{G},\sigma)$, set $u := \text{lca}_T(x,z)$ and let $v_x, v_z \in \text{child}_T(u)$ be the two distinct children of u such that $x \preceq_T v_x$ and $z \preceq_T v_z$. By symmetry, it suffices to show that xz is *u-fp*. Since $\langle zxyz'\rangle$ is a bad quartet, we have $(x,z), (x,z') \in E(\vec{G})$ and thus $\text{lca}_T(x,z') = \text{lca}_T(x,z) = u$. Let $v_{z'} \in \text{child}_T(u)$ be the child of u such that $z' \preceq_T v_{z'}$. Since $\text{lca}_T(x,z') = u$ we have $v_x \neq v_{z'}$. Now, assume for contradiction that $v_z = v_{z'}$, and thus $z' \in L(T(v_z))$. Since $\langle zxyz'\rangle$ is a bad quartet, we have $(z',x) \notin E(\vec{G})$, which implies the existence of a vertex x' with $\sigma(x) = \sigma(x')$ and $\text{lca}_T(x',z') \prec_T \text{lca}_T(x,z') = u$ and therefore, $x' \in L(T(v_z))$. However, this implies that $\text{lca}_T(x',z) \preceq_T v_z \prec_T u = \text{lca}_T(x,z)$, which together with $\sigma(x) = \sigma(x')$ contradicts the fact that xz is an edge in $\vec{G}$. Hence, $v_z \neq v_{z'}$. Therefore, $\sigma(z) = \sigma(z') \in \sigma(L(T(v_z))) \cap \sigma(L(T(v_{z'}))) \neq \emptyset$ for distinct children $v_z, v_{z'} \in \text{child}_T(u)$. By Lemma 6.3, the edge xz is (T,σ)-*fp* and since (T,σ) was chosen arbitrarily, the edge xz is *u-fp*. Moreover, we have shown that v_x, v_z and $v_{z'}$ must be pairwise distinct and thus, (T,σ) is non-binary. $\square$

Fig. 49 shows that *u-fp* edges xy with $\mathcal{S}^{\cap}(x,y) \neq \emptyset$ exist that are neither middle edges of good quartets or first edges of bad quartets. Thus we next consider ugly quartets.

Proposition 6.4. *If $\langle xyx'z\rangle$ is an ugly quartet in a BMG $(\vec{G},\sigma)$, then the edges xy and yx' are* u-fp.

Proof. Consider an ugly quartet $\langle xyx'z\rangle$. Let (T,σ) be an arbitrary tree explaining $(\vec{G},\sigma)$, put $u := \operatorname{lca}_T(x,y)$ and let $v_x, v_y \in \mathsf{child}_T(u)$ be the two distinct children of u such that $x \preceq_T v_x$ and $y \preceq_T v_y$.

Since $x'y$ and xy are edges in $\vec{G}$ we have $\operatorname{lca}_T(x',y) \preceq_T u$. Moreover, Cor. 6.1 implies $\sigma(x') = \sigma(x) \notin \sigma(L(T(v_y)))$ and thus $x' \notin L(T(v_y))$. Therefore, $\operatorname{lca}_T(x',y) = \operatorname{lca}_T(x,y) = u$.

Now consider an arbitrary reconciliation map μ from (T,σ) to some species tree S. The existence of μ is guaranteed by Lemma 6.1. If $x' \notin L(T(v_x))$, then there is a vertex $v_3 \in \mathsf{child}_T(u)$, $v_3 \neq v_x, v_y$ such that $x' \preceq_T v_3$ and $\sigma(x) = \sigma(x') \in \sigma(L(T(v_x))) \cap \sigma(L(T(v_3))) \neq \emptyset$, which by Lemma 6.2 implies $t_\mu(u) = \square$.

Now suppose $x' \in L(T(v_x))$ and recall that $x'z$ is an edge in $\vec{G}$ by assumption. Since $\operatorname{lca}_T(x',z)$ and $\operatorname{lca}_T(x,x')$ are both ancestors of x' they are comparable. If $\operatorname{lca}_T(x',z) \succ_T \operatorname{lca}_T(x,x')$, then $\operatorname{lca}_T(x,z) = \operatorname{lca}_T(x',z)$. Together with the fact that $x'z$ is an edge in $\vec{G}$ but not xz, this implies that there is a $z' \in L[\sigma(z)]$ such that $\operatorname{lca}_T(x,z') \prec_T \operatorname{lca}_T(x,z)$. This in turn implies $\operatorname{lca}_T(x',z') \prec_T \operatorname{lca}_T(x',z)$, which contradicts that $x'z$ is an edge in $\vec{G}$. Therefore, $x' \in L(T(v_x))$ implies $\operatorname{lca}_T(x',z) \preceq_T \operatorname{lca}_T(x,x')$ and $x,x',z \in L(T(v_x))$. Since yz is not an edge in $\vec{G}$ by assumption and Cor. 6.1 implies $\sigma(y) \notin \sigma(L(T(v_x))$, there is a leaf z' with color $\sigma(z') = \sigma(z)$ such that $\operatorname{lca}_T(y,z') \prec_T \operatorname{lca}_T(y,z)$. This is only possible if $z' \in L(T(v_y)) \cap L[\sigma(z)]$. Therefore, $\sigma(z) \in \sigma(L(T(v_x))) \cap \sigma(L(T(v_y)))$ and Lemma 6.2 implies that $t_\mu(u) = \square$.

In summary, $\operatorname{lca}_T(x',y) = \operatorname{lca}_T(x,y) = u$ and $t_\mu(u) = \square$ for every tree explaining $(\vec{G},\sigma)$ and every possible reconciliation map μ from (T,σ) to any species tree. Thus both xy and $x'y$ are *u-fp*. ☐

Proposition 6.5. *Let $(\vec{G},\sigma)$ be a BMG and xy an edge in $\vec{G}$ with $\mathcal{S}^{\cap}(x,y) \neq \emptyset$. Then xy is either the middle edge of some good quartet $\langle zxyz'\rangle$ or the first edge in some ugly quartet $\langle xyx'z\rangle$ or $\langle yxy'z\rangle$.*

Proof. Let (T,σ) be a leaf-colored tree explaining the BMG $(\vec{G},\sigma)$ with symmetric part (G,σ). Let $v_x, v_y \in \mathsf{child}_T(\operatorname{lca}_T(x,y))$ such that $x \preceq_T v_x$ and $y \preceq_T v_y$. Since $\mathcal{S}^{\cap}(x,y) \neq \emptyset$, Lemma 6.6 implies that there is a good quartet $\langle z_1 x^* y^* z_2\rangle$ with $\sigma(x^*) = \sigma(x)$, $\sigma(y^*) = \sigma(y)$, $\sigma(z_1) = \sigma(z_2) = t \in \mathcal{S}^{\cap}(x,y)$, $x^*, z_1 \in L(T(v_x))$ and $y^*, z_2 \in L(T(v_y))$.

If $x = x^*$ and $y = y^*$ we are done. By symmetry it suffices to consider the case $x \neq x^*$. Before we proceed, we consider the (non-)existence of certain edges in the RBMG $G(T,\sigma)$ and the BMG $\vec{G}(T,\sigma)$. By definition of good quartets, we have $x^*z_1, x^*y^*, y^*z_2 \in E(G)$ and Cor. 6.1 implies $\sigma(x), \sigma(y) \notin \mathcal{S}^{\cap}(x,y)$. Hence, $\sigma(x^*) = \sigma(x) \notin \sigma(L(T(v_y)))$ and $\sigma(y^*) = \sigma(y) \notin \sigma(L(T(v_x)))$, and thus $x^*y \in E(G)$ and $xy^* \in E(G)$. Moreover, since $\operatorname{lca}_T(y,z_2) \prec_T \operatorname{lca}_T(y,z_1)$, we have $yz_1 \notin E(G)$. Similarly, $xz_2 \notin E(G)$. However, $\sigma(x) \notin \sigma(L(T(v_y)))$ implies that $\operatorname{lca}_T(z_2,x) = \operatorname{lca}_T(x,y) \preceq_T \operatorname{lca}_T(z_2,x')$ for all $x' \in L[\sigma(x)]$ and thus, $(z_2,x) \in E(\vec{G})$. Similarly, $(z_1,y) \in E(\vec{G})$. Furthermore, we note that neither x and x^* nor y and y^* can be adjacent in G or $\vec{G}$ since $\sigma(x) = \sigma(x^*)$ and $\sigma(y) = \sigma(y^*)$.

If $xz_1 \notin E(G)$, then $\langle xyx^*z_1\rangle$ forms an ugly quartet. Now suppose that $xz_1 \in E(G)$. Assume that there is an edge $yz' \in E(G)$ with $z' \in L(T(v_y)) \cap L[t]$. Then, $\operatorname{lca}_T(x,z_1) \prec_T \operatorname{lca}_T(x,z')$ implies $xz' \notin E(G)$. Moreover, since $\sigma(x) \notin \sigma(L(T(v_y)))$ we have, by similar arguments as above, that $(z',x) \in E(\vec{G})$. Thus, $\langle z'yxz_1\rangle$ forms a good quartet. Finally, if there is no such edge $yz' \in E(G)$ then, in particular, $yz_2 \notin E(G)$ and $y \neq y^*$. In this case, $\langle yxy^*z_2\rangle$ forms an ugly quartet. ☐

The example Fig. 48 shows that the converse of Prop. 6.5 is not true in general.

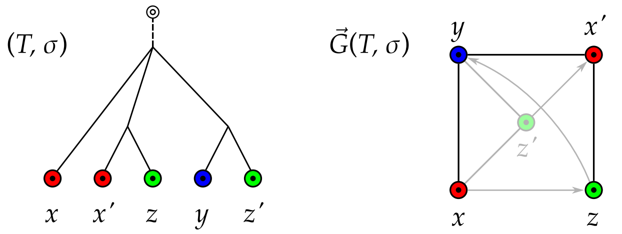


Fig. 48. The edge xy is *u-fp* since it is the first edge of an ugly quartet. However, $\mathcal{S}^{\cap}(x,y)=\emptyset$ and thus, the converse of Prop. 6.5 is not satisfied.

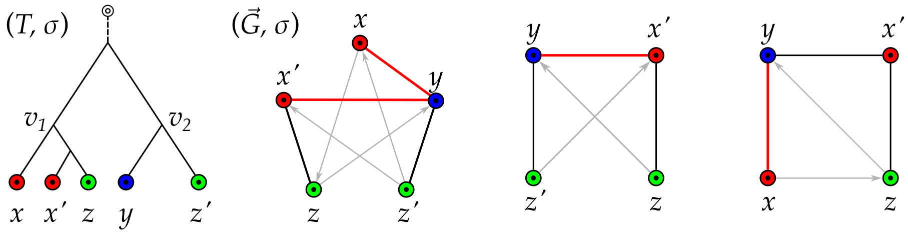


Fig. 49. Example for a (T,σ)-*fp* edge xy in $(\vec{G},\sigma)$ which is not the middle edge of a good quartet, but the first edge in an ugly quartet (right). Note, $(\vec{G},\sigma)$ does not contain bad quartets.

Not surprisingly, quartets are intimately linked to color intersections. We summarize the results of Props. 6.1, 6.2, 6.4 and 6.5 in the following

Corollary 6.3. *Let $(\vec{G},\sigma)$ be a BMG that contains the edge xy. Then, $\mathcal{S}^{\cap}(x,y)\neq\emptyset$ implies that xy is either the middle edge of some good quartet or the first edge of some ugly quartet, which in turn implies that xy is* u-fp*.*

All *u-fp* edges xy with $\mathcal{S}^{\cap}(x,y)\neq\emptyset$ in $(\vec{G},\sigma)$ are therefore completely determined by the middle edges of good quartets and the first edges of ugly quartets. In particular, not all such edges are the middle edge of a good quartet as the example in Fig. 49 shows. Therein, the edge xy must be *u-fp* since $\mathcal{S}^{\cap}(x,y)=\{\sigma(z)\}\neq\emptyset$ (cf. Prop. 6.1). The only good quartet is $\langle zx'yz'\rangle$ identifying $x'y$ as *u-fp*. Moreover, $(\vec{G},\sigma)$ does not contain any bad quartet. The edge xy, on the other hand, is the first edge of the ugly quartet $\langle xyx'z\rangle$.

Furthermore, if an edge xy is the middle edge of a good quartet, then $\mathcal{S}^{\cap}(x,y)\neq\emptyset$. Therefore, only ugly quartets may provide additional information about *u-fp* edges that are not identified with the help of the color-intersection $\mathcal{S}^{\cap}$ (see Fig. 48 for an example). Ugly quartets, however, do not convey all the missing information on *u-fp* edges. The edge xy in the BMG shown in Fig. 50(A), i.e., in an hourglass $[xy \bowtie x'y']$, is *u-fp*, but it is not contained in a good, bad, or ugly quartet.

6.3.2 u-fp *Edges in Hourglasses and Hourglass Chains – The Case* $\mathcal{S}^{\cap}(x,y)=\emptyset$

The case $\mathcal{S}^{\cap}(x,y)\neq\emptyset$ is sufficient to detect the edge xy as *u-fp*. In this section, we turn to the case $\mathcal{S}^{\cap}(x,y)=\emptyset$ and show how to identify further

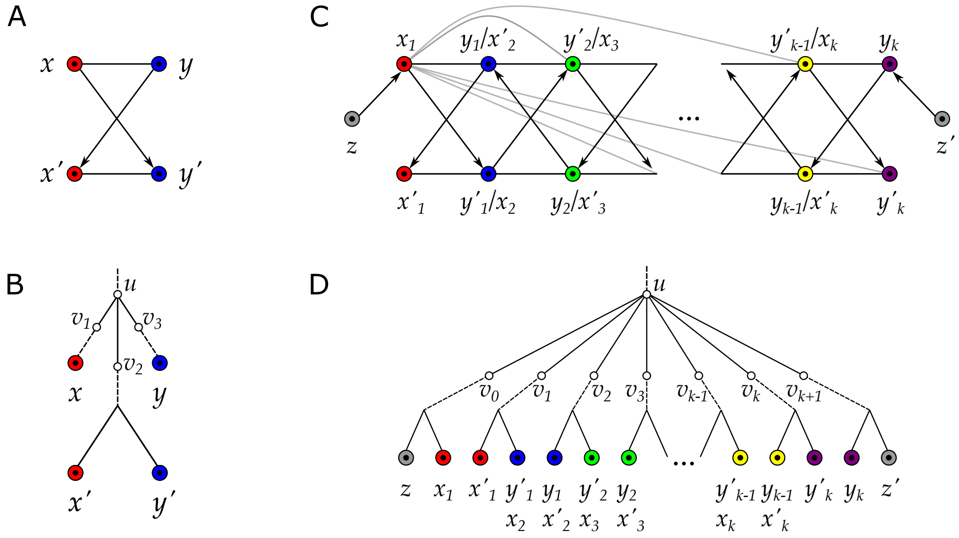


Fig. 50. A: Hourglass. B: The unique tree explaining the hourglass (cf. Lemma 4.15). C: Hourglass chain with left tail z and right tail z' for an odd number of hourglasses in the chain. Edges of the form $x_iy'_j \in E(G)$ are only shown for x_1, the others are omitted. An hourglass chain $\mathfrak{H}$ is a subgraph but not necessarily induced and thus additional arcs may exist. In particular, the elements $e \in \{x_1y_k, zy_k, x_1z', zz'\}$ are not necessarily edges in an hourglass chain. However, whenever they exist, they are *u-fp* (cf. Lemma 6.9). Moreover, each single hourglass in $\mathfrak{H}$ is an induced subgraph of the BMG; by definition, therefore, there are no arcs (z, x'_1) or (z', y'_k). Note, $\sigma(z) \neq \sigma(z')$ is possible. D: Visualization of Lemmas 6.7 and 6.8.

u-fp edges using hourglasses. Recall their definition in Sec. 4.3 which is again illustrated in Fig. 50(A). Hourglasses are not necessarily part of an induced P_4. In particular, an hourglass does not contain an induced P_4.

Lemma 4.15 in Sec. 4.3 shows that hourglasses $[xy \bowtie x'y']$ can be used to identify false-positive edges xy with $\mathcal{S}^{\cap}(x,y) = \emptyset$. More precisely, we have

Proposition 6.6. *If a BMG $(\vec{G}, \sigma)$ contains an hourglass $[xy \bowtie x'y']$, then the edge xy is* u-fp.

Proof. According to Lemma 4.15, every tree (T, σ) that explains $(\vec{G}, \sigma)$ contains a vertex $u \in V^0(T)$ with three distinct children v_1, v_2, and v_3 such that $x \preceq_T v_1$, $\operatorname{lca}_T(x', y') \preceq_T v_2$ and $y \preceq_T v_3$. Thus, $u = \operatorname{lca}_T(x, y)$ and $\sigma(x) \in \sigma(L(T(v_1))) \cap \sigma(L(T(v_2)))$. Hence, we can apply Lemma 6.3 to conclude that xy is (T, σ)*-fp* for every tree that explains $(\vec{G}, \sigma)$. Therefore, the edge xy is *u-fp*. $\square$

Prop. 6.6 implies that there are *u-fp* edges that are not contained in a quartet, see Fig. 50(A). In this example, we have $\mathcal{S}^{\cap}(x,y) = \emptyset$ and no induced P_4. However, as shown in Fig. 50(B), the subtree $T(v_2)$ contains both colors $\sigma(x)$ and $\sigma(y)$ and thus, "bridges" the color sets of the subtrees $T(v_1)$ and $T(v_3)$. Similarly, in the tree (T, σ) in Fig. 50(D), each subtree $T(v_i)$, $1 \leq i \leq k$ "bridges" the color sets of the subtrees $T(v_{i-1})$ and $T(v_{i+1})$. This observation suggests the concept of hourglass chains, a generalization of hourglasses.

Definition 6.8 (Hourglass chain)**.** *An* hourglass chain $\mathfrak{H}$ *in a digraph* $(\vec{G},\sigma)$ *is a sequence of* $k \geq 1$ *hourglasses* $[x_1y_1 \bowtie x'_1y'_1], \dots, [x_ky_k \bowtie x'_ky'_k]$ *such that the following two conditions are satisfied for all* $i \in \{1,\dots,k-1\}$*:*

(H1) $y_i = x'_{i+1}$ *and* $y'_i = x_{i+1}$*, and*

(H2) $x_iy'_j$ *is an edge in* $\vec{G}$ *for all* $j \in \{i+1,\dots,k\}$

A vertex z *is called a* left *(resp.,* right*)* tail *of the hourglass chain* $\mathfrak{H}$ *if it holds that* $(z,x_1) \in E(\vec{G})$ *and* $(z,x'_1) \notin E(\vec{G})$ *(resp.,* $(z,y_k) \in E(\vec{G})$ *and* $(z,y'_k) \notin E(\vec{G})$*). We call* $\mathfrak{H}$ tailed *if it has a left or right tail.*

In contrast to the quartets and the hourglass, an hourglass chain in $(\vec{G},\sigma)$ is not necessarily an induced subgraph.

Observation 6.1. *If* $\mathfrak{H} = [x_1y_1 \bowtie x'_1y'_1], \dots, [x_ky_k \bowtie x'_ky'_k]$ *be an hourglass chain in* $(\vec{G},\sigma)$*, then* $[x_iy_i \bowtie x'_iy'_i], \dots, [x_jy_j \bowtie x'_jy'_j]$ *is an hourglass chain in* $(\vec{G},\sigma)$ *for every* $1 \leq i < j \leq k$*.*

Hourglass chains are "overlapping" hourglasses. The additional condition that $x_iy'_j \in E(G)$ for all $1 \leq i < j \leq k$ ensures that the two pairs x'_k, y'_k and x'_l, y'_l with $k \neq l$ cannot lie in the same subtree below the last common ancestor u which is common to all hourglasses in the chain. More precisely, we have

Lemma 6.7. *Let* $\mathfrak{H} = [x_1y_1 \bowtie x'_1y'_1], \dots, [x_ky_k \bowtie x'_ky'_k]$ *be an hourglass chain in a BMG* $(\vec{G},\sigma)$*. Then, for every tree* (T,σ) *that explains* $(\vec{G},\sigma)$ *there is a vertex* $u \in V^0(T)$ *with pairwise distinct children* $v_0, v_1, \dots, v_k, v_{k+1}$ *such that* $x_1 \in L(T(v_0))$*,* $y_k \in L(T(v_{k+1}))$*, and, for all* $1 \leq i \leq k$*, we have* $x'_i, y'_i \in L(T(v_i))$*.*

Proof. We prove the statement by induction on k. For the base case $k = 1$, observe that the hourglass $[x_1y_1 \bowtie x'_1y'_1]$ together with Lemma 4.15 implies that there is a vertex $u \in V^0(T)$ with pairwise distinct children v_0, v_1 and v_2 such that $x_1 \preceq_T v_0$, $\operatorname{lca}_T(x'_1, y'_1) \preceq_T v_1$ (thus $x'_1, y'_1 \preceq_T v_1$) and $y_1 \preceq_T v_2$.

Now let $k > 1$ and assume that the statement is true for all hourglass chains containing less than k hourglasses. Let $\mathfrak{H} = [x_1y_1 \bowtie x'_1y'_1], \dots, [x_ky_k \bowtie x'_ky'_k]$ be an hourglass chain. By induction hypothesis, for every subsequence $\mathfrak{H}_{i|} := [x_1y_1 \bowtie x'_1y'_1], \dots, [x_iy_i \bowtie x'_iy'_i]$ of $\mathfrak{H}$ with $1 \leq i < k$, which by Obs. 6.1 is again an hourglass chain, the statement is true.

Consider the subsequence $\mathfrak{H}_{i|}$ with $i = k-1$. By assumption, there is a vertex $u \in V^0(T)$ with pairwise distinct children $v_0, v_1, \dots, v_i, v_{i+1}$ such that it holds $x_1 \in L(T(v_0))$, $y_i \in L(T(v_{i+1}))$, and, for all $1 \leq j \leq i$, we have $x'_j, y'_j \in L(T(v_j))$. The hourglass $[x_{i+1}y_{i+1} \bowtie x'_{i+1}y'_{i+1}]$ and Lemma 4.15 imply the existence of a vertex $u' \in V^0(T)$ with pairwise distinct children v'_i, v'_{i+1} and v'_{i+2} such that $x_{i+1} \preceq_T v'_i$, $\operatorname{lca}_T(x'_{i+1}, y'_{i+1}) \preceq_T v'_{i+1}$ and $y_{i+1} \preceq_T v'_{i+2}$. By the definition of hourglass chains, we have $y_i = x'_{i+1}$ and $y'_i = x_{i+1}$. Therefore, $u' = \operatorname{lca}_T(x'_{i+1}, x_{i+1}) = \operatorname{lca}_T(y_i, y'_i) = u$. Since v_i and v'_i are both children of u, $y'_i = x_{i+1}$ and it holds both that $y'_i \preceq_T v_i$ and $x_{i+1} \preceq_T v'_i$, we conclude that $v_i = v'_i$. Similarly, it holds $v_{i+1} = v'_{i+1}$ since $v_{i+1}, v'_{i+1} \in \operatorname{child}_T(u)$ and $y_i = x'_{i+1}$. In particular, we have $v'_{i+2} \neq v'_{i+1} = v_{i+1}$ and $v'_{i+2} \neq v'_i = v_i$. It remains to show that $v'_{i+2} \neq v_j$ for $0 \leq j < i$. Assume, for contradiction, that $v'_{i+2} = v_j$ for some fixed j with $0 \leq j < i$. By assumption, $x_1 \preceq_T v_j$ if $j = 0$, and otherwise, $x_{j+1} = y'_j \preceq_T v_j$. Moreover, since $v'_{i+2} = v_j$, we have $y_{i+1} \preceq_T v_j$. Hence, $\operatorname{lca}_T(x_{j+1}, y_{i+1}) \preceq_T v_j$. Furthermore, since $y'_{i+1} \preceq_T v_{i+1} \neq v_j$, it holds $\operatorname{lca}_T(x_{j+1}, y'_{i+1}) = u \succ_T v_j$. Since $\sigma(y_{i+1}) = \sigma(y'_{i+1})$ by the definition of hourglasses, the latter two arguments contradict $x_{j+1}y'_{i+1} \in E(G)$, which must hold by the definition of hourglass chains. Hence, we can conclude that $v'_{i+2} \neq v_j$ for and

$0 \leq j < i$ and we set $v_{i+2} := v'_{i+2}$. In summary, the statement holds for the hourglass chain $\mathfrak{H}_{i+1|} = \mathfrak{H}$. $\square$

It is straightforward to generalize the latter statement to tailed hourglass chains.

Lemma 6.8. *Let $\mathfrak{H} = [x_1y_1 \bowtie x'_1y'_1], \dots, [x_ky_k \bowtie x'_ky'_k]$ be an hourglass chain with left (resp. right) tail z in a BMG $(\vec{G}, \sigma)$. Then, every tree (T, σ) that explains $(\vec{G}, \sigma)$ contains a vertex $u \in V^0(T)$ with pairwise distinct children $v_0, v_1, \dots, v_k, v_{k+1}$ such that it holds $x_1 \in L(T(v_0))$, $y_k \in L(T(v_{k+1}))$, and, for all $1 \leq i \leq k$, we have $x'_i, y'_i \in L(T(v_i))$. Furthermore, we have $z \preceq_T v_0$ (resp. $z \preceq_T v_{k+1}$).*

Proof. By Lemma 6.7, there is a vertex $u \in V^0(T)$ with pairwise distinct children $v_0, v_1, \dots, v_k, v_{k+1}$ such that it holds $x_1 \in L(T(v_0))$, $y_k \in L(T(v_{k+1}))$, and, for all $1 \leq i \leq k$, we have $x'_i, y'_i \in L(T(v_i))$.

Suppose that z is a left tail of $\mathfrak{H}$. We need to show that $z \preceq_T v_0$. By definition, $(z, x_1) \in E(\vec{G})$, $(z, x'_1) \notin E(\vec{G})$, and $\sigma(x_1) = \sigma(x'_1)$. Therefore, $zx_1|x'_1$ is an informative triple for $(\vec{G}, \sigma)$, and hence $\operatorname{lca}_T(z, x_1) \prec_T \operatorname{lca}_T(z, x'_1) = \operatorname{lca}_T(x_1, x'_1) = u$. Since v_0 is the unique child of u with $x_1 \prec_T v_0$, we can conclude that $\operatorname{lca}_T(z, x_1) \preceq_T v_0$ and thus, $z \preceq_T v_0$.

If z is a right tail of $\mathfrak{H}$, a similar argument using the informative triple $z'y_k|y'_k$, which must be displayed by T because $(z, y_k) \in E(\vec{G})$ and $(z, y'_k) \notin E(\vec{G})$, implies $z \preceq_T v_{k+1}$. $\square$

We are now in the position to show that hourglass chains identify additional *u-fp* edges that are not contained in a single hourglass.

Lemma 6.9. *Let $\mathfrak{H} = [x_1y_1 \bowtie x'_1y'_1], \dots, [x_ky_k \bowtie x'_ky'_k]$ be an hourglass chain in $(\vec{G}, \sigma)$, possibly with a left tail z or a right tail z'. Then every edge $e \in \{x_1y_k, zy_k, x_1z', zz'\} \cap E(G)$ is* u-fp, *where G denotes the symmetric part of $\vec{G}$.*

Proof. Let (T, σ) be an arbitrary tree that explains $(\vec{G}, \sigma)$. By the definition of hourglass chains, we have $k \geq 1$. Hence, the sequence contains at least the hourglass $[x_1y_1 \bowtie x'_1y'_1]$. Since $\mathfrak{H} = [x_1y_1 \bowtie x'_1y'_1], \dots, [x_ky_k \bowtie x'_ky'_k]$ in $\vec{G}(T, \sigma)$, Lemma 6.8 implies the existence of a vertex $u \in V^0(T)$ with pairwise distinct children $v_0, v_1, \dots, v_k, v_{k+1}$ such that it holds $x_1 \in L(T(v_0))$, $y_k \in L(T(v_{k+1}))$, and, for all $1 \leq i \leq k$, we have $x'_i, y'_i \in L(T(v_i))$. Furthermore, this lemma also implies $z \preceq_T v_0$ if z is a left tail of $\mathfrak{H}$, and $z' \preceq_T v_{k+1}$ if z' is a right tail of $\mathfrak{H}$. Note that $\operatorname{lca}_T(x_1, x'_1) = u$, and x_1 and x'_1 lie below distinct children of u. More precisely $x_1 \preceq_T v_0$ and $x'_1 \preceq_T v_1$. Since $\sigma(x_1) = \sigma(x'_1)$, we have $\sigma(L(T(v_0))) \cap \sigma(L(T(v_1))) \neq \emptyset$. Moreover, $\operatorname{lca}_T(a, b) = u$ for every edge $e = ab$ in $\vec{G}$ that coincides with one of x_1y_k, zy_k, x_1z', and zz'. The latter two arguments together with Lemma 6.3 imply that every such edge is (T, σ)*-fp*. Since (T, σ) was chosen arbitrarily, every such edge is also *u-fp*. $\square$

It is important to note that the construction of hourglass chains does not imply that an edge $e \in \{x_1y_k, zy_k, x_1z', zz'\}$ must exist in $(\vec{G}, \sigma)$. Nevertheless, whenever such an edge occurs, it is *u-fp*. We will take a closer look at the properties of hourglass chains in Sec. 6.4.

6.3.3 *Color-set Intersection Graphs*

Our goal is to derive a polynomial-time algorithm to identify all *u-fp* edges in a given BMG. To this end, we now take a closer look at hourglass chains and the trees that explain them. In Fig. 50(D), each subtree $T(v_i)$, $1 \leq i \leq k$, "bridges" the color sets of the subtrees $T(v_{i-1})$ and $T(v_{i+1})$. That is, $\sigma(L(T(v_{i-1}))) \cap \sigma(L(T(v_i)))$ and $\sigma(L(T(v_i))) \cap \sigma(L(T(v_{i+1})))$ are non-empty. This suggests to consider the children of a vertex u as the vertices of a "color-set intersection graph" with edges connecting children with non-empty color-set intersection:

Definition 6.9. *The* color-set intersection graph $\mathfrak{C}_T(u)$ *of an inner vertex u of a leaf-colored gene tree (T, σ) is the undirected graph with vertex set $V := \mathsf{child}_T(u)$ and edge set*

$$E := \{v_1 v_2 \mid v_1, v_2 \in V,\, v_1 \neq v_2 \text{ and } \sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) \neq \emptyset\}.$$

This construction is similar to the definition of intersection graphs e.g. used in [210]. $\mathfrak{C}_T(u)$ can be viewed as a natural generalization of $\mathcal{S}^{\cap}(x, y)$ in the following sense: if $u = \operatorname{lca}_T(x, y)$ is a binary vertex, then $\mathfrak{C}_T(u) = K_2$ *iff* $\mathcal{S}^{\cap}(x, y) \neq \emptyset$ and therefore, $\mathfrak{C}_T(u) = K_1 \cup K_1$ *iff* $\mathcal{S}^{\cap}(x, y) = \emptyset$. In the non-binary case, there is an edge $v_1 v_2$ *iff* $\mathcal{S}^{\cap}(x, y) \neq \emptyset$ for some $x \in L(T(v_1))$ and $y \in L(T(v_2))$.

Shortest paths in the color-set intersection graphs will play an important role in identifying many *u-fp* edges.

Lemma 6.10. *Let v_1 and v_k be two distinct vertices in the same connected component of the color-set intersection graph $\mathfrak{C}_T(u)$ of a leaf-colored gene tree (T, σ), and let $P(v_1, v_k) = (v_1, \dots, v_k)$ be a shortest path in $\mathfrak{C}_T(u)$ connecting v_1 and v_k. Then $\sigma(L(T(v_i))) \cap \sigma(L(T(v_j))) = \emptyset$ for all i and j satisfying $1 \leq i < i + 2 \leq j \leq k$.*

Proof. Assume, for contradiction, that $\sigma(L(T(v_i))) \cap \sigma(L(T(v_j))) \neq \emptyset$ for some i, j with $1 \leq i < i + 2 \leq j \leq k$. Then the edge $v_i v_j$ must be contained in $\mathfrak{C}_T(u)$, contradicting the fact that $P(v_1, v_k)$ is a *shortest* path. ☐

The following lemma establishes a close connection between color-set intersection graphs and hourglass chains.

Lemma 6.11. *Let $(\vec{G}, \sigma)$ be a BMG that is explained by (T, σ) and suppose that $x, y \in L(T)$ are two distinct leaves with $u := \operatorname{lca}_T(x, y)$ and $v_x, v_y \in \mathsf{child}_T(u)$ such that (i) $x \preceq_T v_x$ and $y \preceq_T v_y$, and (ii) there is a shortest path $(v_x = v_0, v_1, \dots, v_k, v_{k+1} = v_y)$ of length at least two in $\mathfrak{C}_T(u)$. Then there is an hourglass chain $\mathfrak{H} = [x_1 y_1 \bowtie x'_1 y'_1], \dots, [x_k y_k \bowtie x'_k y'_k]$ in $(\vec{G}, \sigma)$. In particular, precisely one of the following conditions is satisfied:*

1. *$x_1 = x$ and $y_k = y$;*
2. *$y_k = y$ and $z := x$ is a left tail of $\mathfrak{H}$;*
3. *$x_1 = x$ and $z' := y$ is a right tail of $\mathfrak{H}$; or*
4. *$z := x$ is a left tail and $z' := y$ is a right tail of $\mathfrak{H}$.*

Proof. Lemma 6.10 implies $\mathcal{S}^{\cap}(x,y) = \sigma(L(T(v_x))) \cap \sigma(L(T(v_y))) = \sigma(L(T(v_0))) \cap \sigma(L(T(v_{k+1}))) = \emptyset$. We proceed by showing that the BMG $\vec{G}(T,\sigma)$ contains an hourglass chain $\mathfrak{H} = [x_1y_1 \bowtie x'_1y'_1], \dots, [x_ky_k \bowtie x'_ky'_k]$ possibly with left tail z and right tail z' such that one of the Conditions 1–4 is satisfied.

We first consider the two cases: either (A) $\sigma(x) \in \sigma(L(T(v_1)))$ or (B) $\sigma(x) \notin \sigma(L(T(v_1)))$. In Case (A), we set $x_1 := x$ and $c_0 := \sigma(x)$. In Case (B), we set $z := x$, choose $c_0 \in \sigma(L(T(v_0))) \cap \sigma(L(T(v_1)))$ arbitrarily (note v_0v_1 forms an edge in $\mathfrak{C}_T(u)$ and thus, the latter intersection is non-empty) and we set $x_1 = v$ for some $v \in L(T(v_0)) \cap L[c_0]$ such that $\operatorname{lca}_T(v,x) \preceq_T \operatorname{lca}_T(v',x) \preceq_T v_0$ for all $v' \in L(T(v_0)) \cap L[c_0]$. Clearly, such a vertex v exists. Moreover, $c_0 \neq \sigma(x)$ and we obtain $(x,v) = (z,x_1) \in E(\vec{G})$ as necessary requirement for left tails. In summary, we have in Case (A) $x_1 = x$ and in Case (B) x plays the role of the left tail z and x_1 is some other vertex. Moreover, in both Cases (A) and (B), we have $\sigma(x_1) = c_0 \in \sigma(L(T(v_0))) \cap \sigma(L(T(v_1)))$.

We now consider the "other end" of the hourglass chain, that is, vertex y_k and the possible right tail. Again, we have two cases: either (A′) $\sigma(y) \in \sigma(L(T(v_{k+1})))$ or (B′) $\sigma(y) \notin \sigma(L(T(v_{k+1})))$. In Case (A′), we set $y_k := y$ and $c_k := \sigma(y)$. In Case (B′), we set $z' := y$, and , by similar arguments as in Case (A) and (B), we can choose $c_k \in \sigma(L(T(v_k))) \cap \sigma(L(T(v_{k+1})))$ arbitrarily and set $y_k = w$ for some vertex $w \in L(T(v_{k+1})) \cap L[c_k]$ such that $(y,w) = (z',y_k) \in E(\vec{G})$ as a necessary requirement for right tails. Again, for both cases (A′) and (B′) we have $\sigma(y_k) = c_k \in \sigma(L(T(v_k))) \cap \sigma(L(T(v_{k+1})))$.

We continue by picking an arbitrary color c_i from $\sigma(L(T(v_i))) \cap \sigma(L(T(v_{i+1})))$ for each $1 \leq i < k$. This is possible because $v_iv_{i+1} \in E(\mathfrak{C}_T(u))$, and thus $\sigma(L(T(v_i))) \cap \sigma(L(T(v_{i+1}))) \neq \emptyset$. Note that now $c_i \in \sigma(L(T(v_i))) \cap \sigma(L(T(v_{i+1})))$ holds for all $0 \leq i \leq k$. In particular, the colors $c_0, c_1, \dots, c_k$ are pairwise distinct. To see this, assume, for contradiction, that $c_i = c_j$ for some i,j with $i < j$. Then $c_i \in \sigma(L(T(v_i)))$ and $c_i = c_j \in \sigma(L(T(v_{j+1})))$ which implies $c_i \in \sigma(L(T(v_i))) \cap \sigma(L(T(v_{j+1})))$. This contradicts Lemma 6.10 for $j+1 \geq i+2$.

For each $1 \leq i \leq k$, we have $c_{i-1}, c_i \in \sigma(L(T(v_i)))$. Thus Lemma 4.3 ensures the existence of vertices $x'_i \in L(T(v_i)) \cap L[c_{i-1}]$ and $y'_i \in L(T(v_i)) \cap L[c_i]$ that form an edge $x'_iy'_i$ in $\vec{G}$. By assumption we have $x'_iy'_i \in E(G)$ for all $1 \leq i \leq k$ since $[x_iy_i \bowtie x'_iy'_i]$ is an hourglass. We already set x_1 and y_k. We furthermore set $x_i := y'_{i-1}$ for all $1 < i \leq k$, and $y_i := x'_{i+1}$ for all $1 \leq i < k$. Thus ensures that (H1) in Def. 6.8 is satisfied. Moreover, since $\sigma(x_1) = c_0 = \sigma(x'_1)$ and $\sigma(x_i) = \sigma(y'_{i-1}) = c_{i-1}$ for all $1 < i \leq k$, we have $\sigma(x_i) = c_{i-1} = \sigma(x'_i)$ for all $1 \leq i \leq k$. Similar arguments imply $\sigma(y_i) = c_i = \sigma(y'_i)$ for all $1 \leq i \leq k$.

We next show that the induced subgraph $\vec{G}[x_i, x'_i, y_i, y'_i]$ is an hourglass for $1 \leq i \leq k$ and thus $x_iy'_j$ is an edge in $\vec{G}$ for all $i < j \leq k$. We also know, by construction, that $x'_iy'_i$ is an edge in $\vec{G}$.

Independent of whether x_1 was constructed based on the cases (A) or (B), we have $x_i \preceq_T v_0$ if $i = 1$ and $x_i = y'_{i-1} \preceq_T v_{i-1}$ otherwise. Thus $x_i \preceq_T v_{i-1}$. Likewise, independent of whether y_k was constructed based on the cases (A′) or (B′), we have $y_i \preceq_T v_{k+1}$ if $i = k$ and $y_i = x'_{i+1} \preceq_T v_{i+1}$ otherwise. Thus $y_i \preceq_T v_{i+1}$. In summary, we have $x_i \preceq_T v_{i-1}$; $x'_i, y'_i \preceq_T v_i$; and $y_i \preceq_T v_{i+1}$ for all $i \in \{1, \dots, k\}$. This implies $\operatorname{lca}_T(x_i, y'_i) = \operatorname{lca}_T(x_i, y_i) = \operatorname{lca}_T(x'_i, y_i) = u$. Since $i+1 \geq (i-1)+2$ and $P(v_0, v_{k+1})$ is a shortest path, Lemma 6.10 implies $\sigma(L(T(v_{i-1}))) \cap \sigma(L(T(v_{i+1}))) = \emptyset$.

From $\sigma(x_i) \in \sigma(L(T(v_{i-1})))$ and $\sigma(y_i) \in \sigma(L(T(v_{i+1})))$ we obtain $\sigma(x_i) \notin \sigma(L(T(v_{i+1})))$ and $\sigma(y_i) \notin \sigma(L(T(v_{i-1})))$. Thus, there is no $\tilde{y}$ such that $\sigma(\tilde{y}) = \sigma(y'_i) = \sigma(y_i)$ and $\operatorname{lca}_T(x_i, \tilde{y}) \prec_T u = \operatorname{lca}_T(x_i, y'_i) = \operatorname{lca}_T(x_i, y_i)$, and no $\tilde{x}$ such that $\sigma(\tilde{x}) = \sigma(x'_i) = \sigma(x_i)$ and $\operatorname{lca}_T(y_i, \tilde{x}) \prec_T u = \operatorname{lca}_T(y_i, x'_i) = \operatorname{lca}_T(y_i, x_i)$. Hence, $\vec{G}$ contains the arcs (x_i, y'_i), (x_i, y_i), (y_i, x_i) and (y_i, x'_i). Moreover, x_iy_i is an edge in $\vec{G}$. However, since $\sigma(x'_i) = \sigma(x_i)$ and $\operatorname{lca}_T(x'_i, y'_i) \preceq_T v_i \prec_T u = \operatorname{lca}_T(x_i, y'_i)$ we conclude

$(y'_i, x_i) \notin E(\vec{G})$. Likewise, $\sigma(y'_i) = \sigma(y_i)$ and $\mathrm{lca}_T(x'_i, y'_i) \preceq_T v_i \prec_T u = \mathrm{lca}_T(x'_i, y_i)$ imply that $(x'_i, y_i) \notin E(\vec{G})$. In summary, $\vec{G}[x_i, x'_i, y_i, y'_i] = [x_i y_i \bowtie x'_i y'_i]$ is an hourglass, for all $i \in \{1, \dots, k\}$, and $x_i \preceq_T v_{i-1}$ and $y'_j \preceq_T v_j$ for all $1 \leq i < j \leq k$.

Since $j \geq (i-1) + 2$ and $P(v_0, v_{k+1})$ is a shortest path, Lemma 6.10 implies that $\sigma(L(T(v_{i-1}))) \cap \sigma(L(T(v_j))) = \emptyset$. Thus, there is no $\widetilde{y}$ such that $\sigma(\widetilde{y}) = \sigma(y'_j)$ and $\mathrm{lca}_T(x_i, \widetilde{y}) \prec_T u = \mathrm{lca}_T(x_i, y'_j)$, and no $\widetilde{x}$ such that $\sigma(\widetilde{x}) = \sigma(x_i)$ and $\mathrm{lca}_T(y'_j, \widetilde{x}) \prec_T u = \mathrm{lca}_T(y'_j, x_i)$. This implies that $(x_i, y'_j) \in E(\vec{G})$ and $(y'_j, x_i) \in E(\vec{G})$, respectively. Therefore $x_i y'_j$ is an edge in $\vec{G}$ for $1 \leq i < j \leq k$. In summary, (H2) of in Def. 6.8 is always satisfied.

Hence, if x_1 and y_1 are constructed based on Case (A) and (A'), respectively, we are done.

It remains to show that z and z' are a left and a right tail, resp., of the hourglass chain in Case (B) or (B'). First assume Case (B), and thus $z = x$. We have $z, x_1 \preceq_T v_0$ by construction and $(z, x_1) \in E(\vec{G})$ as shown above. Together with $x'_1 \preceq_T v_1$, this implies that $\mathrm{lca}_T(z, x_1) \preceq_T v_0 \prec_T u = \mathrm{lca}_T(z, x'_1)$. Using $\sigma(x_1) = \sigma(x'_1)$ we therefore obtain $(z, x'_1) \notin E(\vec{G})$. and hence z is a left tail of the constructed hourglass chain. Now assume Case (B'), and thus, $z' = y$. We have $z', y_k \preceq_T v_{k+1}$ and $(z', y_k) \in E(\vec{G})$ by construction. Together with $y'_k \preceq_T v_k$ this implies $\mathrm{lca}_T(z', y_k) \preceq_T v_{k+1} \prec_T u = \mathrm{lca}_T(z', y'_k)$. Using $\sigma(y_k) = \sigma(y'_k)$, we obtain $(z', y'_k) \notin E(\vec{G})$ and hence z' is a right tail of the constructed hourglass chain.

In summary, $\mathfrak{H} = [x_1 y_1 \bowtie x'_1 y'_1], \dots, [x_k y_k \bowtie x'_k y'_k]$ is an hourglass chain, possibly with left tail z and right tail z'. Furthermore, precisely one of the Conditions 1–4 in the statement holds by construction. ☐

6.3.4 *Hug-Edges and No-Hug Graphs*

Definition 6.10. *An edge xy in a vertex-colored digraph $(\vec{G}, \sigma)$ is a* hug-edge *if it satisfies at least one of the following conditions:*

(C1) *xy is the middle edge of a good quartet in $(\vec{G}, \sigma)$;*

(C2) *xy is the first edge of an ugly quartet in $(\vec{G}, \sigma)$; or*

(C3) *there is an hourglass chain $\mathfrak{H} = [x_1 y_1 \bowtie x'_1 y'_1], \dots, [x_k y_k \bowtie x'_k y'_k]$ in $(\vec{G}, \sigma)$, and one of the following cases holds:*

1. *$x_1 = x$ and $y_k = y$;*
2. *$y_k = y$ and $z := x$ is a left tail of $\mathfrak{H}$;*
3. *$x_1 = x$ and $z' := y$ is a right tail of $\mathfrak{H}$; or*
4. *$z := x$ is a left tail and $z' := y$ is a right tail of $\mathfrak{H}$.*

The term **hug**-edge refers to the fact xy is a particular edge of an **h**ourglass-chain, an **u**gly quartet, or a **g**ood quartet. As we shall see later, hug-edges coincide with the *u-fp* edges (cf. Thm. 6.6 below).

Theorem 6.4. *An edge xy in $\vec{G}(T, \sigma)$ with $u := \mathrm{lca}_T(x, y)$, $v_x, v_y \in \mathsf{child}_T(u)$, $x \preceq_T v_x$, and $y \preceq_T v_y$ is a hug-edge if v_x and v_y belong to the same connected component of $\mathfrak{C}_T(u)$. Moreover, every hug-edge is* u-fp.

Proof. We show first that xy satisfies one of the Conditions (C1), (C2), or ((C3), and hence is hug-edge. First, note that $v_x \neq v_y$. Moreover, Lemma 4.4 implies $\sigma(x) \notin \sigma(L(T(v_y)))$ and $\sigma(y) \notin \sigma(L(T(v_x)))$. Since by assumption v_x, v_y belong to the same connected component, there is a shortest path $P := (v_x = v_0, \dots, v_{k+1} = v_y)$ in $\mathfrak{C}_T(u)$. For $k = 0$, $v_x v_y \in E(\mathfrak{C}_T(u))$. This implies $\mathcal{S}^{\cap}(x, y) = \sigma(L(T(v_x))) \cap \sigma(L(T(v_y))) \neq \emptyset$.

By Prop. 6.5, the edge xy is either the middle edge of a good quartet or the first edge of an ugly quartets in $(\vec{G}, \sigma)$. Hence, Condition (C1) or (C2) is satisfied. If $k > 0$, Lemma 6.11 implies Condition (C3).

For each of the three cases we have already shown that xy is *u-fp*: For (C1) Prop. 6.2 applies, for (C2) Prop. 6.4 provides the desired result, and for (C3) we use Lemma 6.9. □

Lemma 6.12. *If the BMG $\vec{G}(T, \sigma)$ contains a hug-edge xy in a BMG $\vec{G}(T, \sigma)$, then there are distinct vertices $v_1, v_2 \in \mathsf{child}_T(\mathsf{lca}_T(x, y))$ such that $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) \neq \emptyset$.*

Proof. Let xy be a hug-edge in the BMG $(\vec{G}, \sigma) = \vec{G}(T, \sigma)$, i.e. one of (C1), (C2), or (C3) applies.

If $e = xy$ satisfies (C1), then xy is the middle edge of a good quartet $\langle zxyz' \rangle$ in $(\vec{G}, \sigma)$. By [104, Lemma 36], there is a vertex $u := \mathsf{lca}_T(x, y, z, z')$ such that $x, z \preceq_T v_1$ and $y, z' \preceq_T$ for some distinct $v_1, v_2 \in \mathsf{child}_T(u)$. Thus, $u = \mathsf{lca}_T(x, y)$. Moreover, since $\sigma(z) = \sigma(z')$, we have $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) \neq \emptyset$ for two distinct vertices $v_1, v_2 \in \mathsf{child}_T(u)$.

If $e = xy$ satisfies (C2), then it is the first edge of some ugly quartet, which w.l.o.g. has the form $\langle xyx'z \rangle$. Re-using the arguments in the proof of Prop. 6.4 shows that there must be two distinct children v_1 and v_2 of vertex $u = \mathsf{lca}_T(x, y)$ such that $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) \neq \emptyset$.

If $e = xy$ satisfies (C3), then there is a (tailed) hourglass chain $\mathfrak{H} = [x_1 y_1 \boxtimes x'_1 y'_1], \dots, [x_k y_k \boxtimes x'_k y'_k]$, $k \geq 1$, in $\vec{G}(T, \sigma)$, such that either $x = x_1$ or $z := x$ is a left tail of $\mathfrak{H}$, and either $y = y_k$ or $z' := y$ is a right tail of $\mathfrak{H}$. In either case, Lemma 6.8 implies $x \preceq_T v_0$ and $y \preceq_T v_{k+1}$. Since x_1 and x'_1 lie below distinct children v_0 and v_1 of vertex $\mathsf{lca}_T(x, y)$ and $\sigma(x_1) = \sigma(x'_1)$ by the definition of hourglasses, it holds that $\sigma(L(T(v_0))) \cap \sigma(L(T(v_1))) \neq \emptyset$.

In each case, therefore, there are distinct vertices $v_1, v_2 \in \mathsf{child}_T(\mathsf{lca}_T(x, y))$ such that $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) \neq \emptyset$. □

The fact that all hug-edges are *u-fp* by Thm. 6.4 suggests to consider the subgraph of a BMG that is left after removing all these unambiguously recognizable false-positive orthology assignments.

Definition 6.11. *Let $(\vec{G}, \sigma)$ be a BMG with symmetric part G and let F be the set of its hug-edges. The* no-hug[1] *graph $\mathbb{NH}(\vec{G}, \sigma)$ is the subgraph of G with vertex set $V(\vec{G})$, coloring σ and edge set $E(G) \setminus F$.*

By Thm. 6.6, $\mathbb{NH}(\vec{G}, \sigma)$ is therefore the subgraph of the underlying RBMG of $(\vec{G}, \sigma)$ that contains all edges that cannot be identified as *u-fp* by using only good quartets, ugly quartets and (tailed) hourglass chains as outlined in Thm. 6.4. Importantly, it contains the orthology graph for every reconciliation map μ as well as the orthology graph induced by the extremal event labeling as subgraphs:

Corollary 6.4. *Let (T, σ) be a leaf-colored tree and μ a reconciliation map from (T, σ) to some species tree S. Then,*

$$\Theta(T, t_\mu) \subseteq \Theta(T, \widehat{t}_T) \subseteq \mathbb{NH}(\vec{G}(T, \sigma)) \subseteq \vec{G}(T, \sigma).$$

---

[1] a good advice in the time of COVID-19

Proof. By Thm. 6.2, $\Theta(T, t_\mu) \subseteq \Theta(T, \widehat{t}_T) \subseteq \vec{G}(T, \sigma)$; and by definition, we have $\mathbb{NH}(\vec{G}(T, \sigma)) \subseteq \vec{G}(T, \sigma)$. Now, let xy be an edge in $\Theta(T, \widehat{t}_T)$ and thus, $\widehat{t}_T(\mathrm{lca}_T(x, y)) = \bullet$. By definition of $\widehat{t}_T$, we have $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) = \emptyset$ for any two distinct $v_1, v_2 \in \mathsf{child}_T(\mathrm{lca}_T(x, y))$. The contraposition of Lemma 6.12 implies that xy is not a hug-edge and thus an edge of $\mathbb{NH}(\vec{G}(T, \sigma))$, which completes the proof. $\square$

The no-hug graph still may contain false-positive orthology assignments, i.e., $\mathbb{NH}(\vec{G}(T, \sigma)) = \Theta(T, t_\mu)$ does not hold in general. As an example, consider the BMG $\vec{G}(T_1, \sigma)$ in Fig. 46. Here, none of the edges xz, $x'z$ and yz are *u-fp* and thus, by Thm. 6.6 also not hug-edges. Hence, they still remain in $\mathbb{NH}(\vec{G}(T_1, \sigma))$. However, these edges are not contained in $\Theta(T_1, \widehat{t}_T)$, since $\widehat{t}_T(\mathrm{lca}_{T_1}(x, x', y, z)) = \square$ and thus, $\Theta(T_1, \widehat{t}_T) \subsetneq \mathbb{NH}(\vec{G}(T_1, \sigma))$. In the following section, we shall see that there are, however, no *u-fp* edges left in the no-hug graph.

6.3.5 *Resolving Least Resolved Trees*

Every BMG $(\vec{G}, \sigma)$ contains all information necessary to determine the trees (T, σ) by which it is explained. Since *u-fp* edges are defined in terms of the explaining trees, every BMG $(\vec{G}, \sigma)$ also contains – at least implicitly – all information needed to identify its *u-fp* edges. Since $(\vec{G}, \sigma)$ is determined by its unique least resolved tree (T^*, σ), the *u-fp* edges must also be determined by (T^*, σ). It is not sufficient for this purpose, however, to find an event labeling t of the vertices of T^*.

To see this, consider for example the "true" history $(\widetilde{T}, \widetilde{t}, \sigma)$ of the BMG $\vec{G}(\widetilde{T}, \sigma)$ as shown in Fig. 51. The unique least resolved tree (T^*, σ) for $\vec{G}(\widetilde{T}, \sigma)$ is obtained by merging the two vertices v_1 and v_2 of $\widetilde{T}$ resulting in the vertex v of T^*. We have $\widetilde{t}(v_1) = \bullet \neq \square = \widetilde{t}(v_2)$. For vertex v and every reconciliation map μ from (T^*, σ) to any species tree S, it must hold that $\mu(v) \in E(S)$ and thus $t^*_\mu(v) = \square$, since v has two children with overlapping color sets and by Lemma 6.2. Thus, the edges cx with $x \in \{a_1, a_2, b_1, b_2\}$ are (T^*, σ)-*fp* although they are not false positives at all. Since speciation and duplication vertices may be merged into the same vertex v of T^*, the least resolved tree T^* in general cannot simply inherit the event labeling from the true gene history, and thus there may not be a "correct" labeling t^* of T^* that provides evidence for all *u-fp* edges.

The example in Fig. 51 shows that the least resolved tree T^* simply may not be "resolved enough". In the following, we therefore describe how the unique least resolved tree can be resolved further to provide more evidence about *u-fp* edges. Eventually, this will lead us to a characterization of the *u-fp* edges. To this end, we need to gain more insights into the structure of redundant edges, i.e., those edges e in T for which (T_e, σ) still explains $\vec{G}(T, \sigma)$.

Since the color sets of distinct subtrees below a speciation vertex cannot overlap by Lemma 6.2, Cor. 4.3 implies that all edges below a speciation vertex are redundant and thus can be contracted. More precisely, we have

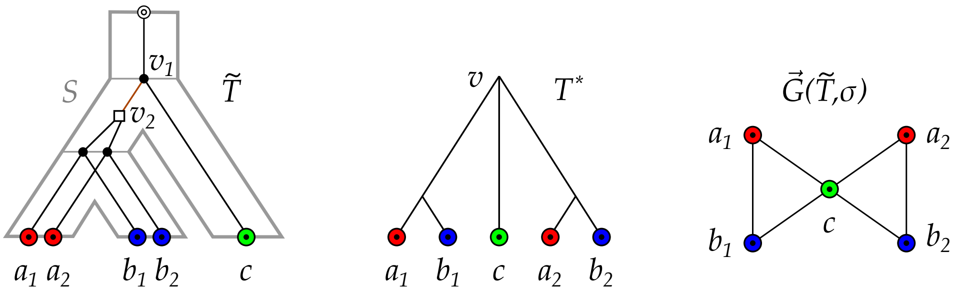

S
v_1
$\widetilde{T}$
v_2
a_1 a_2 b_1 b_2 c
v
T^*
a_1 b_1 c a_2 b_2
$\vec{G}(\widetilde{T},\sigma)$
a_1
a_2
c
b_1
b_2

Fig. 51. The evolutionary scenario (left) shows the event-labeled gene tree $(\widetilde{T}, \widetilde{t}, \sigma)$ embedded into a species tree S. In the least resolved tree (T^*, σ) of $\vec{G}(\widetilde{T}, \sigma)$, the edge v_1v_2 of $\widetilde{T}$ has been contracted into vertex v. The BMG $\vec{G}(\widetilde{T}, \sigma)$ does not contain any *u-fp* edge. *See text for further explanations.*

Observation 6.2. *Let μ be a reconciliation map from (T, σ) to S and assume that there is a vertex $u \in V^0(T)$ such that $\mu(u) \in V^0(S)$ and thus, $t_\mu(u) = \bullet$. Then every inner edge uv of T with $v \in \mathsf{child}_T(u)$ is redundant w.r.t. $\vec{G}(T, \sigma)$. Moreover, if an inner edge uv with $v \in \mathsf{child}_T(u)$ is non-redundant, then u must have two children with overlapping color sets, and hence, $t_\mu(u) = \square$.*

Our goal is to identify those vertices in (T^*, σ) that can be expanded to yield a tree that still explains $\vec{G}(T^*, \sigma)$. To this end, we need to introduce a particular way of "augmenting" a leaf-colored tree.

Definition 6.12. *Let (T, σ) be a leaf-colored tree, u be an inner vertex of T, $\mathfrak{C}_T(u)$ the corresponding color-set intersection graph, and $\mathcal{C}$ the set of connected components of $\mathfrak{C}_T(u)$. Then the tree T_u* augmented at vertex *u is obtained by applying the following editing steps to T:*

- *If $\mathfrak{C}_T(u)$ is connected, do nothing.*
- *Otherwise, for each $C \in \mathcal{C}$ with $|C| > 1$*
  - *introduce a vertex w and attach it as a child of u, i.e., add the edge uw,*
  - *for every element $v_i \in C$, substitute the edge uv_i by the edge wv_i.*

The augmentation step is trivial *if $T_u = T$, in which case we say that* no edit step was performed*.*

An example of an augmentation is shown in Fig. 52.

It is easy to see that the tree T_u obtained by an augmentation of a phylogenetic tree T is again a phylogenetic tree. The augmentation step at vertex u of T is trivial if and only if either $\mathfrak{C}_T(u)$ is connected or all connected components $C \in \mathcal{C}$ are singletons, i.e., $|C| = 1$. If (T_u, σ) is obtained by augmenting (T, σ) at node u, we denote the set of newly introduced vertices by $V_{\neg T} := V(T_u) \setminus V(T)$. Note that $V_{\neg T} = \emptyset$ whenever no edit step was performed.

Since augmentation only inserts vertices between u and its children, it affects neither $L(T(u))$ nor $L(T(v))$ for $v \in \mathsf{child}_T(u)$. As an immediate consequence, we find

Observation 6.3. *Let (T, σ) be a leaf-colored tree, $u \neq v$ two inner vertices of T, $\mathfrak{C}_T(u)$ the corresponding color-set intersection graph, and (T_u, σ) the tree obtained by augmenting T at u. Then $\mathfrak{C}_{T_u}(v) = \mathfrak{C}_T(v)$.*

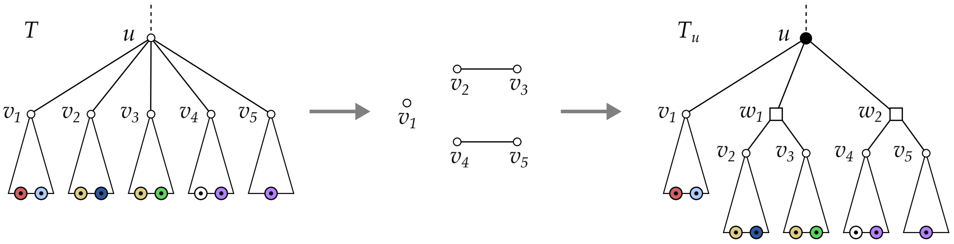


Fig. 52. Left, a (part of a) leaf-colored tree (T,σ). The tree (T_u,σ) on the right is obtained from (T,σ) by augmenting T at vertex u. The color-set intersection graph $\mathfrak{C}_T(u)$ (shown in the middle) has more than one connected component and there are connected components consisting of more than two vertices $v_i \in \mathsf{child}_T(u)$. According to Lemma 6.13, $\sigma(L(T_u(v))) \cap \sigma(L(T_u(v'))) = \emptyset$ for any two distinct vertices $v, v' \in \mathsf{child}_{T_u}(u) = \{v_1, w_1, w_2\}$. By Cor. 4.3, the edges uw_1 and uw_2 are redundant w.r.t. $\vec{G}(T_u,\sigma)$ and thus, both trees explain the same BMG.

Lemma 6.13. *Let (T,σ) be a leaf-colored tree. Let $u \in V^0(T)$ and T_u be the tree after augmenting T at vertex u. If $\mathfrak{C}_T(u)$ is disconnected, then $\sigma(L(T_u(w_1))) \cap \sigma(L(T_u(w_2))) = \emptyset$ for any two distinct vertices $w_1, w_2 \in \mathsf{child}_{T_u}(u)$.*

Proof. By construction, the vertex w_i in T_u, $i = 1,2$, is either a child of u in T or was inserted in the augmentation step. Therefore, the two connected components C_1 and C_2 of $\mathfrak{C}_T(u)$ to which w_1 and w_2 belong are disjoint. Thus $\sigma(L(T(v_i))) \cap \sigma(L(T(v_j))) = \emptyset$ for all $v_i, v_j \in \mathsf{child}_T(u)$ with $v_i \in C_1$ and $v_j \in C_2$ because otherwise there would be an edge v_iv_j in $\mathfrak{C}_T(u)$ and thus, $C_1 = C_2$. Since w_i is either the single vertex in C_i or w_i has as children the vertices of C_i in T_u, $i \in \{1,2\}$, we conclude that $\sigma(L(T_u(w_1))) \cap \sigma(L(T_u(w_2))) = \emptyset$. □

The following result shows that no further edit step can be performed at vertices that have been newly introduced by a previous augmentation step or have already undergone an augmentation.

Lemma 6.14. *Let (T,σ) be a leaf-colored tree, $u \in V^0(T)$, (T_u,σ) the tree obtained by augmenting T at u, and denote by (T_{uw},σ) the tree obtained by augmenting T_u at w. Then $T_{uw} = T_u$ for $w = u$ as well as for all newly introduced vertices, i.e., for all $w \in V_{\neg T} \cup \{u\}$.*

Proof. If $T_u = T$, then $V_{\neg T} = \emptyset$ and thus $T_{uu} = T_u = T$. If $T_u \neq T$, then the definition of the augmentation step at u implies that either $\mathfrak{C}_{T_u}(u)$ is connected or all connected components of $\mathfrak{C}_{T_u}(u)$ are singletons. In either case Lemma 6.13 ensured that augmentation at u leaves T_u unchanged, i.e., $T_{uu} = T_u$. By construction, $\mathfrak{C}_{T_u}(w)$ is connected for $w \in V_{\neg T} \setminus \{u\}$ and thus, we have $T_{uw} = T_u$. □

Hence, a key property of the procedure in Def. 6.12 is that repeated augmentation of the same inner vertex leads to at most one expansion. The tree obtained by augmenting a set of inner vertices of (T,σ) is therefore independent of the order of the augmentation steps.

Definition 6.13 (Augmented tree)**.** *Let (T,σ) be a leaf-colored tree. The* augmented tree *of (T,σ), denoted by $(\mathcal{A}(T),\sigma)$, is obtained by augmenting all inner vertices of (T,σ).*

Algorithm 6: Augmented tree

```
Data: Leaf-colored phylogenetic tree $(T, \sigma)$.
Result: Augmented tree $(\mathcal{A}(T), \sigma)$.
foreach $u \in V^0(T)$ in pre-order do
    Compute $\mathfrak{C}_T(u)$.
    $\mathcal{C} \leftarrow$ set of connected components of $\mathfrak{C}_T(u)$
    if $|\mathcal{C}| > 1$ then
        foreach $C \in \mathcal{C}$ such that $|C| > 1$ do
            Introduce a vertex $w$ and the edge $uw$.
            foreach $v_i \in C$ do
                Remove the edge $uv_i$.
                Add the edge $wv_i$.
            end
        end
end
```

The following result ensures that such an augmented tree is unique.

Lemma 6.15. *For every leaf-colored tree (T, σ), there is a unique tree $(\mathcal{A}(T), \sigma)$ obtained from (T, σ) by repeated application of augmentation steps until only trivial augmentation steps remain. The tree $(\mathcal{A}(T), \sigma)$ is computed by Alg. 6.*

Proof. Lemma 6.14 together with Obs. 6.3 implies that (i) every vertex u in T can be non-trivially augmented at most once, (ii) the newly introduced vertices cannot be non-trivially augmented at all, and (iii) augmentation of two distinct inner vertices of T yields the same result irrespective of the order of the augmentation steps. Thus, $(\mathcal{A}(T), \sigma)$ is unique. The correctness of Alg. 6 now follows immediately. $\square$

Lemma 6.16. *Alg. 6 with input $T = (V, E)$ and σ runs in $O(|V|^2|\mathcal{S}|)$ time and $O(|V|^2)$ space, where $\mathcal{S} = \sigma(L(T))$ is the set of species under consideration.*

Proof. Assigning the color set $L(T(u))$ to each u requires $O(|V||\mathcal{S}|)$ time, where $|\mathcal{S}| < |V|$. The total effort to construct all $\mathfrak{C}_T(u)$ is bounded by $O(|V|^2|\mathcal{S}|)$, corresponding to comparing the color sets of all pairs of vertices of T. The total size of all color-set intersection graphs in $O(|V|^2)$. Computation of the connected components is linear in the size of the graph, which also bounds the editing effort for each u, implying the claim. $\square$

We finally show that augmentation preserves the best match relation, i.e., it does not affect the underlying BMG.

Proposition 6.7. *For every leaf-colored tree (T, σ), it holds $\vec{G}(T, \sigma) = \vec{G}(\mathcal{A}(T), \sigma)$.*

Proof. Let $u \in V^0(T)$ and T_u be the tree after augmenting T at vertex u. Put $A := \{uw \mid w \in V_{\neg T}\}$ and note that all edges of T_u in A are inner edges. Now consider $e \in A$. Since $w \in V_{\neg T}$, an edit step was performed to obtain w and thus, $|\mathcal{C}| > 1$ in $\mathfrak{C}_T(u)$. Lemma 6.13 and $|\mathcal{C}| > 1$ imply that for any $v' \in \mathsf{child}_{T_u}(u)$ with $v' \neq w$ we have $\sigma(L(T_u(v'))) \cap \sigma(L(T_u(w))) = \emptyset$. Thus, Cor. 4.3 implies that the edge uw is redundant in (T_u, σ) w.r.t. $\vec{G}(T, \sigma)$.

Denoting by T_{u_A} the tree obtained from T_u by contraction of all edges in A, we obtain $(T, \sigma) = (T_{u_A}, \sigma)$. Lemma 4.8 now implies $\vec{G}(T_u, \sigma) = \vec{G}(T_{u_A}, \sigma) = \vec{G}(T, \sigma)$ for every augmentation step. By Lemma 6.15, we can repeat this argument for every augmentation in the arbitrary order in which $\vec{G}(\mathcal{A}(T), \sigma)$ is obtained from $\vec{G}(T, \sigma)$, and thus $\vec{G}(\mathcal{A}(T), \sigma) = \vec{G}(T, \sigma)$. $\square$

6.3.6 *Extremal Labeling of Augmented Trees*

While the least resolved tree in general cannot support an event labeling that properly reflects the underlying true history of a gene family, we shall see here that the augmented tree $(\mathcal{A}(T),\sigma)$ does feature sufficient resolution. To this end, we investigate the extremal event labeling of $(\mathcal{A}(T),\sigma)$.

Lemma 6.17. *Let $\widehat{t} := \widehat{t}_{\mathcal{A}(T)}$ be the extremal event labeling of the augmented tree $(\mathcal{A}(T),\sigma)$ obtained from (T,σ) and let u be some vertex of $\mathcal{A}(T)$. Then it holds $\widehat{t}(u) = \square$ if and only if $\mathfrak{C}_{\mathcal{A}(T)}(u)$ is connected.*

Proof. By the definitions of the extremal event labeling and $\mathfrak{C}_{\mathcal{A}(T)}(u)$, the 'if'-direction is clear. Now suppose that $\widehat{t}(u) = \square$. There are two possibilities:
(1) $u \in V^0(T)$. If $\mathfrak{C}_T(u)$ is connected, then $\mathfrak{C}_{\mathcal{A}(T)}(u) = \mathfrak{C}_T(u)$. Otherwise, Lemma 6.13 implies that $\sigma(L(\mathcal{A}(T)(w_1))) \cap \sigma(L(\mathcal{A}(T)(w_2))) = \emptyset$ for all $w_1, w_2 \in \mathsf{child}_{\mathcal{A}(T)}(u)$, thus the definition of the extremal event labeling implies $\widehat{t}(u) \neq \square$, a contradiction.
(2) $u \in V_{\neg T}$, i.e., u is newly created by augmenting some $u' \in V^0(T)$, hence $\mathfrak{C}_T(u)$ is connected and, by Obs. 6.3 and Lemma 6.14, $\mathfrak{C}_{\mathcal{A}(T)}(u)$ is connected. □

For later reference, we need the following

Lemma 6.18. *Let $(\vec{G},\sigma)$ be a BMG, (T^*,σ) its least resolved tree, and $\widehat{t} := \widehat{t}_{\mathcal{A}(T^*)}$ the extremal event labeling of the augmented tree $(\mathcal{A}(T^*),\sigma)$. Then, $(\mathcal{A}(T^*),\widehat{t},\sigma)$ does not contain adjacent speciation vertices, i.e., if $\widehat{t}(u) = \bullet$ for a vertex u of $\mathcal{A}(T^*)$, then $\widehat{t}(v) = \square$ for any of its non-leaf children $v \in \mathsf{child}_{\mathcal{A}(T^*)}(u) \setminus L(\mathcal{A}(T^*))$.*

Proof. Set $\mathcal{A} := \mathcal{A}(T^*)$ and note that, by Prop. 6.7, $(\mathcal{A},\sigma)$ explains $(\vec{G},\sigma)$. Assume, for contradiction, that there is an inner edge uv in $\mathcal{A}$ with $v \prec_{\mathcal{A}} u$ such that $\widehat{t}(u) = \widehat{t}(v) = \bullet$. By the definition of the extremal event labeling $\widehat{t}$, we have $\sigma(L(\mathcal{A}(v))) \cap \sigma(L(\mathcal{A}(v'))) = \emptyset$ for any $v' \in \mathsf{child}_{\mathcal{A}}(u) \setminus \{v\}$. Together with Cor. 4.3 this implies that uv is redundant for $(\vec{G},\sigma)$, and hence, not an edge in the least resolved tree (T^*,σ). Now consider the augmentation in which the edge uv, and thus vertex v was created; resulting in a tree (T',σ). By the definition of augmenting (Def. 6.12), it clearly holds that $\mathfrak{C}_{T'}(v)$ is connected. By Lemma 6.14, the edges adjacent to v do not change in any subsequent augmentation. Thus $\mathfrak{C}_{\mathcal{A}}(v)$ must be connected as well. Lemma 6.17 now implies that $\widehat{t}(v) = \square$; a contradiction. □

Lemma 6.19. *Let $(\vec{G},\sigma)$ be a BMG and (T^*,σ) its unique least resolved tree. Moreover, let $\widehat{t} := \widehat{t}_{\mathcal{A}(T^*)}$ be the extremal event labeling of the augmented tree $(\mathcal{A}(T^*),\sigma)$. Then, $\Theta(\mathcal{A}(T^*),\widehat{t}) \subseteq \vec{G}$.*

Proof. Since (T^*,σ) explains $(\vec{G},\sigma)$, we have $(\vec{G},\sigma) = \vec{G}(T^*,\sigma)$. By Prop. 6.7, we have $\vec{G}(T^*,\sigma) = \vec{G}(\mathcal{A}(T^*),\sigma)$. Let xy be an edge in $\Theta(\mathcal{A}(T^*),\widehat{t})$. By definition, $\widehat{t}(\mathsf{lca}_{\mathcal{A}(T^*)}(u)) = \bullet$ where $u := \mathsf{lca}_{\mathcal{A}(T^*)}(x,y)$. By definition of the extremal event labeling, $\sigma(L(\mathcal{A}(T^*)(v_1))) \cap \sigma(L(\mathcal{A}(T^*)(v_2))) = \emptyset$ for all two distinct vertices $v_1, v_2 \in \mathsf{child}_{\mathcal{A}(T^*)}(u)$. The latter is true, in particular, for the two children $v_x, v_y \in \mathsf{child}_{\mathcal{A}(T^*)}(u)$ with $x \preceq_{\mathcal{A}(T^*)} v_x$ and $y \preceq_{\mathcal{A}(T^*)} v_y$. Therefore, $\sigma(x) \notin \sigma(L(\mathcal{A}(T^*)(v_y)))$ and $\sigma(y) \notin \sigma(L(\mathcal{A}(T^*)(v_x)))$. We conclude that x and y are reciprocal best matches in $\mathcal{A}(T^*)$. Finally, $(\vec{G},\sigma) = \vec{G}(\mathcal{A}(T^*),\sigma)$ implies that xy is an edge in $\vec{G}$. □

Now we are in the position to prove the main results of this chapter.

Theorem 6.5. *Let* $(\vec{G},\sigma)$ *be a BMG,* (T^*,σ) *its unique least resolved tree, and* $\widehat{t} := \widehat{t}_{\mathcal{A}(T^*)}$ *the extremal event labeling of the augmented tree* $(\mathcal{A}(T^*),\sigma)$*. Then* $(\Theta(\mathcal{A}(T^*),\widehat{t}),\sigma) = \mathbb{NH}(\vec{G},\sigma)$*.*

Proof. Let (G,σ) be the symmetric part of $(\vec{G} = (V,E),\sigma)$. For simplicity, we write $G_\Theta := \Theta(\mathcal{A}(T^*),\widehat{t})$ and $G_{\mathbb{NH}} := (V, E(\mathbb{NH}(\vec{G},\sigma)))$. Recall that, by definition, $G_{\mathbb{NH}} \subseteq G$ and, by Lemma 6.19, $G_\Theta \subseteq \vec{G}$. Finally, as G contains only edges of $\vec{G}$, we have $G_\Theta \subseteq G$. Let $F := E(G) \setminus E(G_{\mathbb{NH}})$ be the set of all edges of G that are hug-edges, and let $F' := E(G) \setminus E(G_\Theta)$ be the set of all edges in G that do not form orthologous pairs. Since $G_{\mathbb{NH}}, G_\Theta \subseteq G$ it suffices to verify that $F = F'$ in order to show that $(G_\Theta,\sigma) = (G_{\mathbb{NH}},\sigma)$.

Assume $e = xy \in F'$. Hence, $xy \notin E(G_\Theta)$ and therefore, $\widehat{t}(u) = \square$ where $u := \operatorname{lca}_{\mathcal{A}(T^*)}(x,y)$. By Lemma 6.17, $\mathfrak{C}_{\mathcal{A}(T^*)}(u)$ has exactly one connected component. This together with Thm. 6.4 implies that xy is a hug-edge and thus, $xy \in F$, and hence $F' \subseteq F$.

Assume $e = xy \in F$ is a hug-edge. Assume, for contradiction, that $e \notin F'$ and thus, $\widehat{t}(u) = \bullet$ where $u := \operatorname{lca}_{\mathcal{A}(T^*)}(x,y)$. By definition of the extremal event labeling, it must therefore hold that $\sigma(L(\mathcal{A}(T^*)(v_1))) \cap \sigma(L(\mathcal{A}(T^*)(v_2))) = \emptyset$ for any two distinct vertices $v_1, v_2 \in \mathsf{child}_{\mathcal{A}(T^*)}(u)$. By Prop. 6.7, $(\mathcal{A}(T^*),\sigma)$ explains $(\vec{G},\sigma)$. This together with Lemma 6.12 implies that there are two distinct vertices $v_1, v_2 \in \mathsf{child}_{\mathcal{A}(T^*)}(u)$ such that $\sigma(L(\mathcal{A}(T^*)(v_1))) \cap \sigma(L(\mathcal{A}(T^*)(v_2))) \neq \emptyset$; a contradiction. Therefore, $e \in F'$, and hence $F \subseteq F'$. □

Theorem 6.6. *An edge* xy *in a BMG* $(\vec{G},\sigma)$ *is* u-fp *if and only if* xy *is a hug-edge of* $(\vec{G},\sigma)$*.*

Proof. Let $(\vec{G},\sigma)$ be a BMG, (T^*,σ) its unique least resolved tree, and $\widehat{t} := \widehat{t}_{\mathcal{A}(T^*)}$ the extremal event labeling of the augmented tree $(\mathcal{A}(T^*),\sigma)$. As shown in the proof of Thm. 6.5, every edge xy of of the symmetric part G that is not a hug-edge satisfies $xy \in E(G_\Theta)$ and therefore $\widehat{t}(u) = \bullet$, where $u := \operatorname{lca}_{\mathcal{A}(T^*)}(x,y)$. Lemma 6.3 implies that e is not $(\mathcal{A}(T^*),\sigma)$*-fp* and thus, in particular, not *u-fp*. That is, all edges in $(G_\Theta,\sigma) = (G_{\mathbb{NH}},\sigma)$ are non-*u-fp* edges. Moreover, Thm. 6.4 implies that all hug-edges in $E(G) \setminus E(G_{\mathbb{NH}})$ are *u-fp*. Since $(G_{\mathbb{NH}},\sigma)$ does not contain *u-fp* edges, all *u-fp* edges must also be hug-edges, which completes the proof. □

Since $(\Theta(\mathcal{A}(T^*),\widehat{t}),\sigma) = \mathbb{NH}(\vec{G},\sigma)$ is the subgraph of the underlying RBMG of $(\vec{G},\sigma)$ that does not contain any *u-fp* edges (cf. Def. 6.11 and Thm. 6.6), the set of all *u-fp* edges can readily be obtained by comparing the edges of $(\vec{G},\sigma)$ with the edges in the orthology graph obtained from $(\mathcal{A}(T^*),\widehat{t})$. Since only *u-fp* edges have been removed to obtain $(\Theta(\mathcal{A}(T^*),\widehat{t}),\sigma)$ and since $(\mathcal{A}(T^*),\sigma)$ still explains $(\vec{G},\sigma)$, the graph $(\Theta(\mathcal{A}(T^*),\widehat{t}),\sigma)$ is, in the sense of an unambiguous editing, the best estimate of the orthology relation that we can make by solely utilizing the structural information of a given BMG $(\vec{G},\sigma)$. Note, Thm. 6.1 implies that $\mathbb{NH}(\vec{G},\sigma)$ must, in particular, be a cograph.

Orthology prediction tools intended for large data sets often do not attempt to infer the orthology graph, but instead are content with summarizing the information as *clusters of orthologous groups* (COGs) in an empirically estimated RBMG [251, 309]. Formally, this amounts to editing the BMG to a set of disjoint cliques. The example in Fig. 51 shows that this approach can destroy correct orthology information: the BMG $(\vec{G},\sigma)$ does not contain *u-fp* edges and thus, it is the closest orthology graph. However, $(\vec{G},\sigma)$ is not the disjoint union of cliques.

Interestingly, bad quartets turn out to be redundant for the identification of *u-fp* edges in the sense that every *u-fp* edge in a bad quartet appears as a *u-fp* edge in a good quartet, an ugly quartet, or an hourglass chain. At present, we do not know whether hourglass chains in a colored digraph $(\vec{G},\sigma)$ can be found efficiently. However, the identification of *u-fp* edges does not require the explicit enumeration of hourglass chains. More precisely, since $(\Theta(\mathcal{A}(T^*),\widehat{t}),\sigma) = \mathbb{NH}(\vec{G},\sigma)$, the computation of $\mathbb{NH}(\vec{G},\sigma)$ can be achieved in polynomial time and avoids the need to find the hourglass chains of $(\vec{G},\sigma)$. In fact, the effort is dominated by computing the least resolved tree (T^*,σ) for a given BMG.

Theorem 6.7. *For a given BMG $(\vec{G},\sigma)$, the set of all* u-fp *edges can be computed in $O(|L|^3|\mathcal{S}|)$ time, where $L = V(\vec{G})$ and $\mathcal{S} = \sigma(L(T))$ is the set of species under consideration.*

Proof. Given a BMG $(\vec{G},\sigma)$, its least resolved tree (T^*,σ) can be computed in $O(|L|^3|\mathcal{S}|)$ time (cf. [102, Sec. 5]). The augmented tree $(\mathcal{A}(T^*),\sigma)$ can be obtained from (T^*,σ) in $O(|L|^2|\mathcal{S}|)$ time according to Lemma 6.16. The extremal event labeling $\widehat{t}$ can be obtained from the connectivity information on the $\mathfrak{C}_{\mathcal{A}(T^*)}(u)$ in linear time. Computing $(\Theta(\mathcal{A}(T^*),\widehat{t}),\sigma) = \mathbb{NH}(\vec{G},\sigma)$ then only requires evaluation of $\mathrm{lca}_{\mathcal{A}(T^*)}(x,y)$, which can be achieved in polynomial time in $O(|L|^2)$ as described in [102, Sec. 5]). $\square$

As argued in [102, Sec. 5], the number of genes between different species will be comparable in practical applications, i.e., $O(\ell) = O(|L|/|\mathcal{S}|)$ with $\ell = \max_{s\in\mathcal{S}} |L[s]|$. In this case, the running time to compute (T^*,σ) reduces to $O(|L|^3/|\mathcal{S}|)$ and we obtain an overall running time to compute the set of all *u-fp* edges of $O(|L|^3/|\mathcal{S}| + |L|^2|\mathcal{S}|)$. Thms. 6.5 and 6.7 imply that we do not need to find induced quartets and hourglasses explicitly, nor do we need to identify the hourglass chains. Instead, it is more efficient to compute the least resolved tree (T^*,σ), its augmented tree $(\mathcal{A}(T^*),\sigma)$, and the corresponding extremal event labeling $\widehat{t}$.

6.3.7 *Additional Unidentified False Positives*

Deletion of all *u-fp* edges is necessary to obtain an orthology relation without false positives. It is not sufficient, however, since $\mathbb{NH}(\vec{G},\sigma)$ may contain additional false-positive orthology assignments. In order to construct an example, we consider for a BMG $(\vec{G},\sigma)$ the set $\mathfrak{T}$ of all trees (T,t,σ) for which $\mathbb{NH}(\vec{G},\sigma) = (\Theta(T,t),\sigma)$. The example in Fig. 53 shows that it may be the case that none of the trees $(T,t,\sigma) \in \mathfrak{T}$ admits a reconciliation map μ to any species tree such that $t_\mu = t$.

For an event-labeled, leaf-colored tree (T,t,σ), we consider the triple set

$$\mathfrak{S}(T,t,\sigma) = \{\sigma(a)\sigma(b)|\sigma(c)\colon ab|c \leq T;\ t(\mathrm{lca}_T(a,b,c)) = \bullet;\\ \sigma(a),\sigma(b),\sigma(c) \text{ pairwise distinct}\}. \tag{38}$$

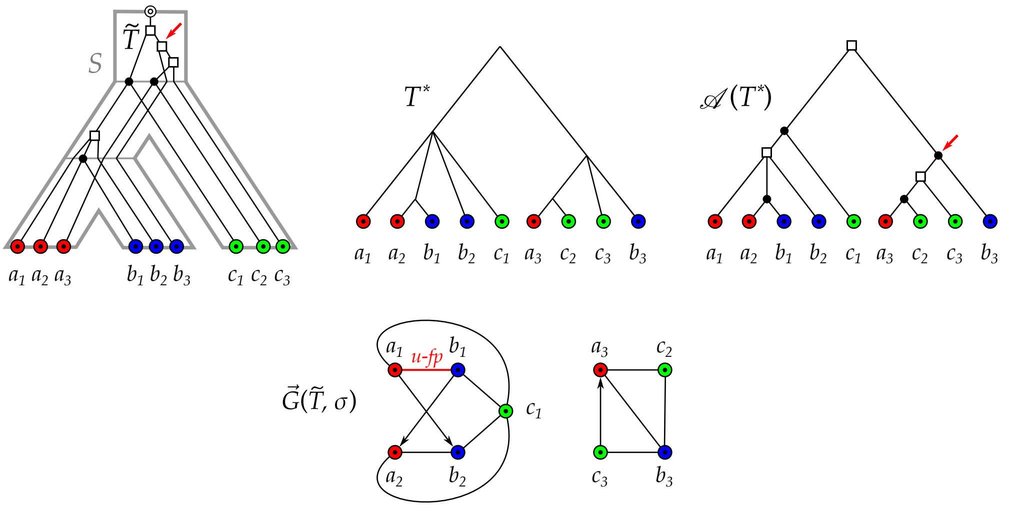


Fig. 53. An evolutionary scenario (left) with a no-hug graph $\mathbb{NH}(\vec{G},\sigma)$ that still contains false-positive edges. Deletion of the highlighted *u-fp* edge a_1b_1 for $\vec{G}(\widetilde{T},\sigma)$ yields $\mathbb{NH}(\vec{G},\sigma) = (\Theta(\mathcal{A}(T^*),\widehat{t}),\sigma)$ and thus, an orthology graph. However, none of its cotrees can be reconciled with any species tree since each of them contains the contradictory species triples $\sigma(a_1)\sigma(b_1)|\sigma(c_1)$ and $\sigma(a_1)\sigma(c_1)|\sigma(b_1)$ (see e.g. [130, 140]). Note, the trees $(\widetilde{T},\widetilde{t})$ and $(\mathcal{A}(T^*),\widehat{t})$ differ in the event label marked by the arrows, resulting in the three additional *fp* edges a_3b_3, c_2b_3 and c_3b_3 in $\mathbb{NH}(\vec{G},\sigma)$.

Moreover, we will need the following characterization of biologically plausible event-labeled gene trees:

Theorem 6.8. [130, 140] *There is a species tree S together with a reconciliation map μ from (T,t,σ) to S such that $t_\mu = t$ if and only if $\mathfrak{S}(T,t,\sigma)$ is consistent. In this case, every species tree S that displays $\mathfrak{S}(T,t,\sigma)$ can be reconciled with (T,t,σ). Moreover, there is a polynomial-time algorithm that determines whether a species tree for (T,t,σ) exists, and if so, returns a species tree S together with a reconciliation map $\mu : T \to S$.*

Throughout this section, we are only concerned with the extremal event labeling $\widehat{t}_{\mathcal{A}(T^*)}$ of the augmented trees $(\mathcal{A}(T^*),\sigma)$ of least resolved trees (T^*,σ). For brevity, we simply write $\widehat{t}$. For a BMG $(\vec{G},\sigma)$, we consider the set of trees

$$\mathfrak{T} := \left\{(T,t,\sigma) \mid \mathbb{NH}(\vec{G},\sigma) = (\Theta(T,t),\sigma)\right\}. \tag{39}$$

An orthology relation $\mathbb{NH}(\vec{G},\sigma)$ obtained from a BMG $(\vec{G},\sigma)$ by removing all of its *u-fp* edges is biologically feasible only if there is an event-labeled gene tree $(T,t,\sigma) \in \mathfrak{T}$ that can be reconciled with some species tree. To show that this condition can be tested in polynomial time, we first need a technical result.

Lemma 6.20. *Let $(\vec{G},\sigma)$ be a BMG with LRT (T^*,σ), and let $\mathfrak{T}$ be be given by Eq. (39). If $ab|c$ is displayed by $\mathcal{A}(T^*)$ and $\widehat{t}(\mathrm{lca}_{\mathcal{A}(T^*)}(a,b,c)) = \bullet$, then $ab|c$ is also displayed by every tree $(T,t,\sigma) \in \mathfrak{T}$ and $t(\mathrm{lca}_T(a,b,c)) = \bullet$.*

Proof. Suppose that $ab|c$ is displayed by $\mathcal{A}(T^*)$ and $\widehat{t}(\mathrm{lca}_{\mathcal{A}(T^*)}(a,b,c)) = \bullet$. Thm. 6.5 implies $(\Theta(\mathcal{A}(T^*),\widehat{t}),\sigma) = \mathbb{NH}(\vec{G},\sigma)$. Thus $\mathbb{NH}(\vec{G},\sigma)$ is a cograph by Thm. 6.1. Let

(T',t',σ) be a least resolved tree for the cograph $\mathbb{NH}(\vec{G},\sigma)$. Clearly, $(T',t',\sigma) \in \mathfrak{T}$. This tree is unique and any other tree in $\mathfrak{T}$ must be a refinement of (T',t',σ) [30, 57]. We proceed with showing that (1) $t'(\mathrm{lca}_{T'}(a,b,c)) = \bullet$ and (2) $ab|c$ is displayed by T'.

In order to show (1), assume for contradiction that $t'(\mathrm{lca}_{T'}(a,b,c)) = \square$ and note that $(T',t',\sigma) \in \mathfrak{T}$ implies $\mathbb{NH}(\vec{G},\sigma) = (\Theta(T',t'),\sigma)$. Since $\widehat{t}(\mathrm{lca}_{\mathcal{A}(T^*)}(a,b,c)) = \bullet$ and $ab|c \le \mathcal{A}(T^*)$, the induced subgraph of $\mathbb{NH}(\vec{G},\sigma)$ on $\{a,b,c\}$ contains at least the two edges ac and bc. However, if $t'(\mathrm{lca}_{T'}(a,b,c)) = \square$, then this induced subgraph can contain at most one edge; a contradiction. Hence, $t'(\mathrm{lca}_{T'}(a,b,c)) = \bullet$.

Next, we show (2). Since $\mathcal{A}(T^*)$ displays $ab|c$ and T' is obtained from $\mathcal{A}(T^*)$ by a series of edge contractions, T' can neither display $ac|b$ nor $bc|a$, thus either $ab|c \le T'$ or $\mathrm{lca}_{T'}(a,b) = \mathrm{lca}_{T'}(a,b,c)$. By Lemma 6.18, $(\mathcal{A}(T^*),\widehat{t})$ does not contain adjacent (consecutive) speciation vertices. Therefore and since $\mathcal{A}(T^*)$ displays $ab|c$, the path from $\mathrm{lca}_{\mathcal{A}(T^*)}(a,b,c)$ to $\mathrm{lca}_{\mathcal{A}(T^*)}(a,b)$ in $\mathcal{A}(T^*)$ must contain at least one duplication vertex. Since T' can be obtained from $\mathcal{A}(T^*)$ by contracting all edges uv in $\mathcal{A}(T^*)$ with $\widehat{t}(u) = \widehat{t}(v)$ [30, 57], the path from $\mathrm{lca}_{T'}(a,b,c)$ to $\mathrm{lca}_{T'}(a,b)$ in T' must contain at least one duplication vertex. Together with $t'(\mathrm{lca}_{T'}(a,b,c)) = \bullet$ this implies $\mathrm{lca}_{T'}(a,b) \neq \mathrm{lca}_{T'}(a,b,c)$, and hence, $ab|c$ is displayed by T'.

Since every tree $(T,t,\sigma) \in \mathfrak{T}$ is a refinement of (T',t',σ), the triple $ab|c$ is also displayed by T. Finally, since $\mathbb{NH}(\vec{G},\sigma) = (\Theta(T,t),\sigma)$ for every tree $(T,t,\sigma) \in \mathfrak{T}$, we can re-use the arguments from the proof of Statement (1) to conclude that $t(\mathrm{lca}_T(a,b,c)) = \bullet$. □

Lemma 6.21. *Let $(\vec{G},\sigma)$ be a BMG with LRT (T^*,σ) and let $\mathfrak{T}$ be given by Eq. (39). Then, the following statements are equivalent:*

1. *There is no reconciliation map μ from $(\mathcal{A}(T^*),\widehat{t},\sigma)$ to any species tree such that $t_\mu = \widehat{t}$.*
2. *For all trees (T,t,σ) in $\mathfrak{T}$ there is no reconciliation map μ from (T,t,σ) to any species tree such that $t_\mu = t$.*

In particular, Condition (1) can be verified in polynomial time.

Proof. First note that $(\mathcal{A}(T^*),\widehat{t},\sigma) \in \mathfrak{T}$ since, by Thm. 6.5, $(\Theta(\mathcal{A}(T^*),\widehat{t}),\sigma) = \mathbb{NH}(\vec{G},\sigma)$. Hence, Statement (2) implies (1).

For the converse, let $ab|c$ be displayed by $\mathcal{A}(T^*)$ where $\sigma(a) = A$, $\sigma(b) = B$, $\sigma(c) = C$ are pairwise distinct, and $\widehat{t}(\mathrm{lca}_{\mathcal{A}(T^*)}(a,b,c)) = \bullet$. By definition, $AB|C \in \mathfrak{S}(\mathcal{A}(T^*),\widehat{t},\sigma)$. Lemma 6.20 implies that $ab|c$ is also displayed by every tree $(T,t,\sigma) \in \mathfrak{T}$ and $t(\mathrm{lca}_T(a,b,c)) = \bullet$. Therefore, we have $\mathfrak{S}(\mathcal{A}(T^*),\widehat{t},\sigma) \subseteq \mathfrak{S}(T,t,\sigma)$ for all $(T,t,\sigma) \in \mathfrak{T}$. Now suppose that Condition (1) holds. Then, by Thm. 6.8, $\mathfrak{S}(\mathcal{A}(T^*),\widehat{t},\sigma)$ is not consistent. Thus, $\mathfrak{S}(T,t,\sigma)$ must be inconsistent as well for every tree $(T,t,\sigma) \in \mathfrak{T}$. Together with Thm. 6.8, this implies Condition (2).

Using the arguments in the proof of Thm. 6.7 and Thm. 6.8 we find that Condition (1) can be verified in polynomial time by checking whether $\mathfrak{S}(\mathcal{A}(T^*),\widehat{t},\sigma)$ is inconsistent. □

Hence, the augmented tree $(\mathcal{A}(T^*),\widehat{t},\sigma)$ is sufficient to test in polynomial time whether or not $\mathfrak{T}$ contains a reconcilable tree, i.e., whether or not the cograph $\mathbb{NH}(\vec{G},\sigma)$ is a biologically feasible orthology relation for $(\vec{G},\sigma)$. In the negative case, we have clear evidence that $\mathbb{NH}(\vec{G},\sigma)$ still contains a false-positive edge and thus must be edited further. This type of false-positive orthology assignments is the topic of ongoing work.

Now consider again a true evolutionary scenario $(\widetilde{T},\widetilde{t},\sigma)$. In contrast to the LRT (T^*,σ) of a BMG $\vec{G}(\widetilde{T},\sigma)$, its augmented tree $\mathcal{A}(T^*)$ is not necessarily displayed by the true gene tree of the underlying evolutionary scenario. As an example, consider the scenario in Fig. 51. Augmenting the only multifurcation in this case further resolves the root of T^* and thus yields a tree that is not displayed by $\widetilde{T}$. Hence, we advocate the augmented tree endowed with the corresponding extremal event labeling $(\mathcal{A}(T^*),\widehat{t},\sigma)$ primarily as convenient tool to identify false-positive orthology assignments. Whether or not $(\mathcal{A}(T^*),\widehat{t},\sigma)$ is a plausible representation of the gene phylogeny depends on whether it admits a reconciliation of the (phylogenetically correct) species tree. As discussed above, this is not always the case. The following result, however, shows that $(\mathcal{A}(T^*),\widehat{t},\sigma)$ is informative in an important special case.

Lemma 6.22. *Let (T,t,σ) be an event-labeled tree explaining the BMG $(\vec{G},\sigma)$, and let (T^*,σ) be the least resolved tree of $(\vec{G},\sigma)$. If $(\Theta(T,t),\sigma) = \mathbb{NH}(\vec{G},\sigma)$, then $\mathcal{A}(T^*)$ is displayed by T.*

Proof. Let $\mathfrak{T}$ be the set of trees corresponding to $(\vec{G},\sigma)$ as given by Eq. (39). First note that $(T,t,\sigma) \in \mathfrak{T}$ and that (T^*,σ) is displayed by (T,σ) [cf. 102, Thm. 8]. Now consider the set $r(\mathcal{A}(T^*))$ of all triples displayed by $\mathcal{A}(T^*)$. For any triple $ab|c \in r(\mathcal{A}(T^*))$, there are exactly two cases: (a) $\widehat{t}(u) = \bullet$ and (b) $\widehat{t}(u) = \square$, where $u := \mathrm{lca}_{\mathcal{A}(T^*)}(a,b,c)$.

In Case (a), Lemma 6.20 together with $(T,t,\sigma) \in \mathfrak{T}$ immediately implies that $ab|c$ is also displayed by T.

In Case (b), we have $\widehat{t}(u) = \square$. Consider the child $v \in \mathsf{child}_{\mathcal{A}(T^*)}(u)$ with $a,b \prec_{\mathcal{A}(T^*)} v$. Assume, for contradiction, that v is not a vertex in T^*, i.e., it was newly created by augmenting a vertex u'. We have $u' = u$ by Lemma 6.14 since u' cannot be (non-trivially) augmented any further. Since $\mathcal{A}(T^*)$ does not depend on the order of augmentation steps, we may assume w.l.o.g. that v was created in the first augmentation step; resulting in the augmented tree T_u. Def. 6.12 implies that $\mathfrak{C}_T(u)$ is disconnected. Together with Lemma 6.13, this implies $\sigma(L(T_u(w_1))) \cap \sigma(L(T_u(w_2))) = \emptyset$ for any two distinct vertices $w_1, w_2 \in \mathsf{child}_{T_u}(u)$. This must still hold for $(\mathcal{A}(T^*),\sigma)$ since the edges uw, where $w \in \mathsf{child}_{T_u}(u)$ correspond to the vertices that have been newly introduced in the first augmentation step, do not change in any subsequent augmentation due to Lemma 6.14. The definition of the extremal event labeling now implies $\widehat{t}(u) = \bullet$; a contradiction. Therefore, we conclude that v is a vertex in T^*, and in particular, $a,b \in L(T^*(v))$ and $c \notin L(T^*(v))$, which in turn implies that $ab|c$ is displayed by T^*. From $T^* \leq T$ we finally conclude that T also displays $ab|c$. Denoting by $r(T)$ the set of all triples displayed by T we therefore have $r(\mathcal{A}(T^*)) \subseteq r(T)$. Finally, we apply Thm. 1 in Bryant and Steel [37] to conclude that $\mathcal{A}(T^*)$ is displayed by T. $\square$

Lemma 6.22 guarantees that $\mathcal{A}(T^*)$ is displayed by the true gene tree $\widetilde{T}$ whenever $\mathbb{NH}(\vec{G},\sigma)$ equals the true orthology relation. In a practical workflow, it can be checked efficiently whether there is evidence for additional false-positive edges because $\mathfrak{T}$ contains no reconcilable tree. If this is not the case, then it is likely that $\mathbb{NH}(\vec{G},\sigma)$ equals the true orthology relation. In this case, $\widetilde{T}$ also displays the unique discriminating cotree of $\mathbb{NH}(\vec{G},\sigma)$.

One has to keep in mind, however, that it is not possible to find a mathematical guarantee for $\mathbb{NH}(\vec{G},\sigma)$ to be the true orthology relation, because it cannot be ruled out that the true scenario contains unwitnessed duplications that are compensated by additional gene losses. Additional evidence, such as the assumption of a molecular clock or synteny may be used to resolve situations such as the complementary loss shown in Fig. 44. In the extreme case, it is logically possible for every BMG that, in the true scenario, all inner vertices of the gene tree predate the root of the species tree, resulting in a true orthology graph without any edges [103, 115, 238]. This comes at the expense of reassigning events as duplications and usually requires a large number of loss events as an explanation. Of course, this is extremely unlikely for real data. Therefore, the question arises whether one can characterize and efficiently compute the minimal relabelings? In the general case, a further refinement of $\mathcal{A}(T^*)$ may be sufficient. Is a refinement of speciation nodes sufficient, or are there in general speciation nodes in $(\mathcal{A}(T^*), \widehat{t})$ that need to be refined into separate speciation and duplication events?

6.4 QUARTETS, HOURGLASSES, AND THE STRUCTURE OF RECIPROCAL BEST MATCH GRAPHS

The characterization of *u-fp* edges is in a way surprising when compared to previous results on the structure of RBMGs [103, 104], which were focused on P_4s and quartets. The expected connection between good and ugly quartets and *u-fp* edges is captured by Cor. 6.3. However, Prop. 6.6 implies that there are also *u-fp* edges entirely unrelated to quartets and thus induced P_4s. In this section, we aim to close this gap in our understanding.

6.4.1 *Hourglass-Free BMGs*

We start with an important special case for which quartets are sufficient. By Thm. 6.3 and Prop. 6.3, this is the case for binary-explainable, i.e., hourglass-free (Prop. 4.8), BMGs. In particular, an hourglass-free BMG also does not contain an hourglass chain.

Geiß et al. [104] found that a certain type of colored 6-cycles is an important characteristic of RBMGs with a "complicated" structure that can only be explained by multifurcating trees. Let us write $\langle x_1 x_2 \dots x_k \rangle$ for an induced cycle C_k with edges $x_i x_{i+1}$, $1 \leq i \leq k-1$, and $x_k x_1$ in the symmetric part G of $\vec{G}$. We say that $(\vec{G}, \sigma)$ contains a *hexagon* if the corresponding RBMG (G, σ) contains an induced $C_6 = \langle x_1 x_2 \dots x_6 \rangle$ such that any three consecutive vertices of C_6 have pairwise distinct colors, i.e., $\sigma(x_i) = \sigma(x_i + 3)$, $1 \leq i \leq 3$. Since hexagons contain P_4s and, by [104, Lemma 32], any P_4 is either a good or a bad quartet, there are exactly two possible induced subgraphs spanned by a hexagon $C_6 = \langle x_1 x_2 \dots x_6 \rangle$, which are shown in Fig. 54. A digraph $(\vec{G}, \sigma)$ is *hexagon-free* if it does not contain a hexagon.

Lemma 6.23. *Every hourglass-free BMG $(\vec{G}, \sigma)$ is hexagon-free.*

Proof. By Prop. 4.8, every hourglass-free BMG $(\vec{G}, \sigma)$ can be explained by a binary tree. Lemma 9 in [103] implies that hexagons can only be explained by non-binary trees. Hence, $(\vec{G}, \sigma)$ must be hexagon-free. ☐

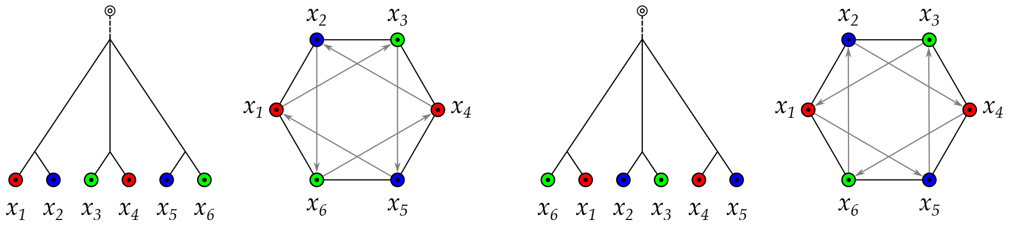

x_1 x_2 x_3 x_4 x_5 x_6
x_2 x_3 x_1 x_4 x_6 x_5
x_6 x_1 x_2 x_3 x_4 x_5
x_2 x_3 x_1 x_4 x_6 x_5

Fig. 54. Two examples of trees whose BMGs $\vec{G}(T,\sigma)$ contain a hexagon $\langle x_1x_2x_3x_4x_5x_6\rangle$. There are exactly two distinct possibilities for the placement of the non-symmetric arcs in the subgraph of the BMG induced by the hexagon.

Clearly, the converse of Lemma 6.23 is not always satisfied, since, by Obs. 4.5, an hourglass is a BMG without hexagons.

A very useful observation in previous work is the fact that every 3-colored vertex induced subgraph of an RBMG (G,σ) is again an RBMG [104, Thm. 7]. Furthermore, the connected components (C,σ) of every 3-colored vertex induced subgraph of (G,σ) belong to precisely one of the three types [104, Thm. 5]:

Type (A) (C,σ) contains a K_3 on three colors but no induced P_4.

Type (B) (C,σ) contains an induced P_4 on three colors whose endpoints have the same color, but no induced cycle C_n on $n \geq 5$ vertices.

Type (C) (C,σ) contains a hexagon.

The graphs for which all such 3-colored connected components are of Type (A) are exactly the RBMGs that are cographs, or *co-RBMGs* for short [104, Thm. 8 and Remark 2]. Together with Lemma 6.23, this classification immediately implies

Corollary 6.5. *Let $(\vec{G},\sigma)$ be an hourglass-free BMG. Then its symmetric part (G,σ) is either a co-RBMG or it contains an induced P_4 on three colors whose endpoints have the same color, but no induced cycle C_n on $n \geq 5$ vertices.*

As a consequence of Thm. 6.6, all *u-fp* edges in an hourglass-free BMG are identified by the good and ugly quartets, which are 3-colored by construction. In hourglass-free BMGs, it is indeed sufficient to consider only the 3-colored P_4s to identify all *u-fp* edges and thus, to obtain an orthology graph, even though the BMG may also contain 4-colored P_4s. Since hourglasses can only appear in BMGs that require multifurcations for their explanation (cf. Lemma 4.15), the case of hourglass-free BMGs is the most relevant for practical applications.

Since all *u-fp* edges in an hourglass-free BMG are contained in quartets, it is also easy to identify the hourglass-free BMGs that are already orthology graphs.

Corollary 6.6. *Let $(\vec{G},\sigma)$ be an hourglass-free BMG. Then its symmetric part (G,σ) is a co-RBMG if and only if there are no* u-fp *edges in $(\vec{G},\sigma)$.*

Proof. Since (G, σ) is a cograph, it contains no induced P_4s and thus, $(\vec{G}, \sigma)$ contains no good or ugly quartets. By Thm. 6.6, all hug-edges are determined by hourglass chains and good or ugly quartets. Since none of them is contained in $(\vec{G}, \sigma)$, it also does not contain *u-fp* edges. Conversely, suppose that $(\vec{G}, \sigma)$ contains no *u-fp* edges. Then, by Thm. 6.5, $(G, \sigma) = \mathbb{NH}(\vec{G}, \sigma)$ is an orthology graph and thus, by Thm. 6.1, a cograph. $\square$

6.4.2 u-fp *Edges in Hourglass Chains*

The situation is much more complicated in the presence of hourglasses. We start by providing sufficient conditions for *u-fp* edges that are identified by hourglass chains.

Proposition 6.8. *Let* $\mathfrak{H} = [x_1y_1 \mathrel{\text{⤨}} x_1'y_1'], \dots, [x_ky_k \mathrel{\text{⤨}} x_k'y_k']$ *be an hourglass chain in* $(\vec{G}, \sigma)$*, possibly with a left tail* z *or a right tail* z'*. Then, an edge in* $\vec{G}$ *is* u-fp *if it is contained in the set*

$$
\begin{aligned}
F = & \{x_iy_j \mid 1 \leq i \leq j \leq k\} \cup \{zz'\} \cup \{zy_i, x_iz', zy_i', x_i'z' \mid 1 \leq i \leq k\} \\
& \cup \{x_ix_{j+1} \mid 1 \leq i < j < k\} \cup \{y_iy_{j+1} \mid 1 \leq i < j < k\} \\
& \cup \{x_1'y_i', x_1'y_i \mid 2 \leq i \leq k\} \cup \{x_iy_k', x_i'y_k' \mid 1 \leq i \leq k-1\} \\
& \cup \{x_1'z, x_1'z', y_k'z, y_k'z'\}
\end{aligned}
$$

Proof. Let (T, σ) be an arbitrary tree that explains $(\vec{G}, \sigma)$. By analogous arguments as in the proof of Lemma 6.9 and by Lemma 6.8, there is a vertex $u \in V^0(T)$ with pairwise distinct children $v_0, v_1, \dots, v_k, v_{k+1}$ such that it holds $x_1 \in L(T(v_0))$, $y_k \in L(T(v_{k+1}))$ and, for all $1 \leq i \leq k$, we have $x_i', y_i' \in L(T(v_i))$. Since $x_{i+1} = y_i'$ and $x_{i+1}' = y_i$ by definition of hourglass chains, it is an easy task to verify that for all edges $e = ab \in F$ the vertices a and b are located below distinct children of u and thus, $\operatorname{lca}_T(a, b) = u$ for all such edges. As argued in the proof of Lemma 6.9, we have $\sigma(L(T(v_0))) \cap \sigma(L(T(v_1))) \neq \emptyset$. The latter arguments together with Lemma 6.3 imply that every edge in F is *u-fp*. $\square$

Figs. 50 and 55 furthermore show that hourglass chains identify false-positive edges that are not associated with quartets in the BMG: The BMG in Fig. 50(A) has the *u-fp* edge xy, and the BMG in Fig. 55(B) contains the *u-fp* edges x_1y_2, x_1z' and $x_1'z'$. A careful investigation shows that these edges are either not even part of an induced P_4 (such as xy in Fig. 50 and $x_1'z'$ in Fig. 55), or at least not identifiable as *u-fp* via good, bad or ugly quartets according to Props. 6.2, 6.3 and 6.4, as it is the case for x_1y_2 and x_1z' in Fig. 55.

The observation that there are false-positive edges which are not associated with quartets limits the use of cograph editing in the context of orthology detection, at least in the case of gene trees with polytomies: On one hand, an RBMG can be a cograph and still contain *u-fp* edges and, on the other hand, there are examples where deletion of the *u-fp* edge identified by quartets (and thus, by induced P_4s) is not sufficient to arrive at a cograph, see Fig. 55.

6.4.3 *Four-Colored* P_4*s*

Geiß et al. [104, Thm. 8] established that the RBMG (G, σ) is a co-RBMG, i.e., a cograph, if and only if every subgraph induced on three colors is a

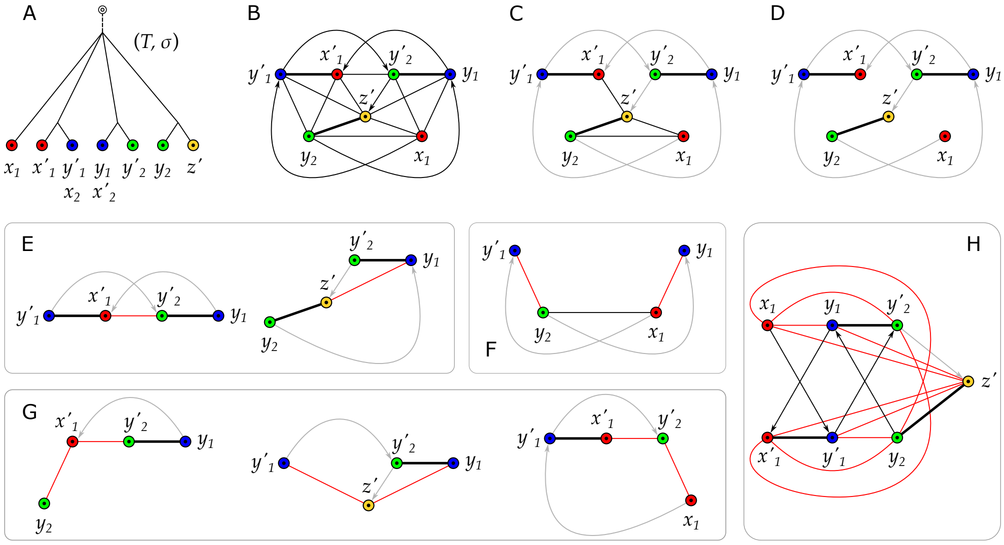


Fig. 55. The (non-binary) tree (T,σ) in Panel (A) explains the BMG $(\vec{G},\sigma)$ in Panel (B), which contains several induced P_4s and an hourglass chain of length $k=2$ with right tail z'. Edges that are not (T,σ)-*fp* (and thus not *u-fp*) are shown as thick lines. Thin edges correspond to those that can be identified as *u-fp* by the subgraphs in (E–H), where they are highlighted in red. (C) The graph after deletion of all edges that can be identified by good, bad and ugly quartets according to Props. 6.2, 6.3, and 6.4. Note that it contains the induced P_4s $\langle y_1'x_1'z'y_2\rangle$ and $\langle y_1'x_1'z'x_1\rangle$, which were not induced subgraphs of the original BMG in (B). Its symmetric part (H,σ) differs from $\mathbb{NH}(\vec{G},\sigma)$ (cf. Def. 6.11) since it still contains *u-fp* edges. (D) The BMG after deletion of all *u-fp* edges. Its symmetric part, comprising the thick edges, is $\mathbb{NH}(\vec{G},\sigma)$. (E) The two good quartets. (F) The single bad quartet. (G) Examples for ugly quartets that cover the remaining *u-fp* edges that are identifiable via quartets. Panel (H) shows the BMG $(\vec{G},\sigma)$ in a different layout that highlights the hourglass chain with right tail z'. All edges that are *u-fp* according to Prop. 6.8 are in red. To identify the *u-fp* edges in $(\vec{G},\sigma)$, only the subgraphs in Panel (E), (G) and (H) are necessary (cf. Def. 6.10 and Thm. 6.5).

cograph. Therefore, if (G,σ) contains an induced 4-colored P_4, it also contains an induced 3-colored P_4. For hourglass-free BMGs $(\vec{G},\sigma)$ it is clear that a 4-colored P_4 always overlaps with a 3-colored P_4: In this case $\mathbb{NH}(\vec{G},\sigma)$ is obtained by deleting middle edges of good quartets and first edges of ugly quartets. Since $\mathbb{NH}(\vec{G},\sigma)$ is a cograph, there is no P_4 left, and thus at least one edge of any 4-colored P_4 was among the deleted edges. It is natural to ask whether this is true for BMGs in general. Fig. 56 shows that good and ugly quartets are not sufficient on their own: there are 4-colored P_4s that do not overlap with the middle edge of a good quartet or the first edge of an ugly quartet. On the other hand, it is clear that at least one of its edges is *u-fp*. This does not imply, however, that the *u-fp* edges in a 4-colored P_4 are also edges of 3-colored P_4s.

Still, in the context of cograph-editing approaches, it is of interest whether the 3-colored P_4s are sufficient. In the following, we provide an affirmative answer.

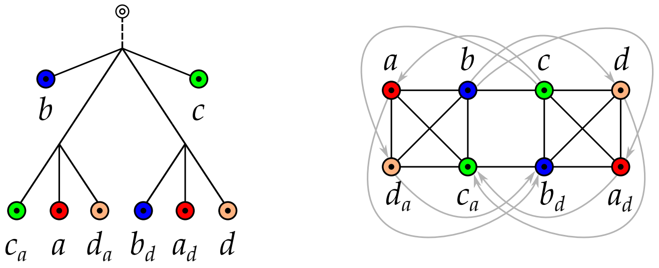


Fig. 56. The symmetric part of the BMG $(\vec{G},\sigma)$ contains the 4-colored induced P_4 $\langle abcd\rangle$. None of its edges is the middle edge of a good quartet or the first edge of an ugly quartet. According to Lemma 6.24, there is the bad quartet $\langle abca_d\rangle$ that contains as first edge the edge ab.

Lemma 6.24. *Let $(\vec{G},\sigma)$ be a BMG and $\mathcal{P}$ a 4-colored induced P_4 in the symmetric part of $(\vec{G},\sigma)$. Then at least one of the edges of $\mathcal{P}$ is either the middle edge of some good quartet or the first edge of a bad or ugly quartet in $(\vec{G},\sigma)$.*

Proof. Let (T,σ) be an arbitrary tree that explains $(\vec{G},\sigma)$ and suppose that $\mathcal{P} := \langle abcd\rangle$ is a 4-colored induced P_4 in the symmetric part (G,σ).

If one of the edges ab, bc, or cd of $\mathcal{P}$ is the middle edge of some good quartet or the first edge of some ugly quartet, then we are done. Hence, we assume in the following that this is not the case and show that at least one of the edges of $\mathcal{P}$ is the first edge in a bad quartet.

By contraposition of Prop. 6.5, we have $\mathcal{S}^{\cap}(a,b)=\emptyset$, $\mathcal{S}^{\cap}(b,c)=\emptyset$ and $\mathcal{S}^{\cap}(c,d)=\emptyset$. We set $v:=\operatorname{lca}_T(b,c)$ with children $v_b,v_c\in\mathsf{child}_T(v)$ such that $b\preceq_T v_b$ and $c\preceq_T v_c$, and $w:=\operatorname{lca}_T(a,b)$ with children $w_a,w_b\in\mathsf{child}_T(w)$ such that $a\preceq_T w_a$ and $b\preceq_T w_b$. Note, that v, v_b, w, and w_b are pairwise comparable, since they are all ancestors of b.

We show that $w=v$. Assume, for contradiction, that (i) $w\prec_T v$ or (ii) $v\prec_T w$. In Case (i), we have $w_a\prec_T w\preceq_T v_b$ and thus, $\sigma(a)\in\sigma(L(T(v_b)))$. Hence, as $\mathcal{S}^{\cap}(b,c)=\emptyset$, it must hold that $\sigma(a)\notin\sigma(L(T(v_c)))$ and $\sigma(c)\notin\sigma(L(T(v_b)))$. Lemma 4.4 implies $ac\in E(G)$. But then $\mathcal{P}$ is not an induced P_4; a contradiction. In Case (ii), we have $v_c\preceq_T v\preceq_T w_b$ and thus, $\sigma(c)\in\sigma(L(T(w_b)))$. Since $\mathcal{S}^{\cap}(a,b)=\emptyset$ we thus have $\sigma(c)\notin\sigma(L(T(w_a)))$ and $\sigma(a)\notin\sigma(L(T(w_b)))$. By Lemma 4.4, $ac\in E(G)$; again a contradiction. Thus $w=v$. Analogous arguments can be used to establish $\operatorname{lca}_T(c,d)=v$. We therefore have $v=\operatorname{lca}_T(a,b)=\operatorname{lca}_T(b,c)=\operatorname{lca}_T(c,d)$. In the following v_x denotes the child of v with $x\preceq_T v_x$ for $x\in\{a,b,c,d\}$. Note, $v_a\neq v_b$, $v_b\neq v_c$ and $v_c\neq v_d$.

We next show that v_a, v_b, v_c, and v_d are pairwise distinct. Fist, assume for contradiction that $v_a=v_c$. Together with $\mathcal{S}^{\cap}(c,d)=\emptyset$, this assumption implies that $\sigma(a)\notin\sigma(L(T(v_d)))$ and $\sigma(d)\notin\sigma(L(T(v_c)))$. By Lemma 4.4, $ad\in E(G)$, contradicting the assumption that $\mathcal{P}$ is an induced P_4. Hence, $v_a\neq v_c$. By symmetry of $\mathcal{P}$, we can use similar arguments to conclude that $v_b\neq v_d$. Finally, assume for contradiction that $v_a=v_d$. Then, $\sigma(d)\in\sigma(L(T(v_a)))$. Hence, $\mathcal{S}^{\cap}(a,b)=\emptyset$ implies that $\sigma(d)\notin\sigma(L(T(v_b)))$ and $\sigma(b)\notin\sigma(L(T(v_d)))$. Again Lemma 4.4 implies $bd\in E(G)$; a contradiction. In summary, v_a, v_b, v_c, and v_d must be pairwise distinct.

We claim $\sigma(c)\in\sigma(L(T(v_a)))$. Since $ad\notin E(G)$ and $\operatorname{lca}_T(a,d)=v$, Lemma 4.4 implies that $\sigma(a)\in\sigma(L(T(v_d)))$ or $\sigma(d)\in\sigma(L(T(v_a)))$. By symmetry of $\mathcal{P}$, we can w.l.o.g. assume that $\sigma(a)\in\sigma(L(T(v_d)))$ and thus, there is a vertex $a_d\in L(T(v_d))$ with $\sigma(a_d)=\sigma(a)$. In this case, $\mathcal{S}^{\cap}(c,d)=\emptyset$ implies that $\sigma(a)\notin\sigma(L(T(v_c)))$. This together with $ac\notin E(G)$ and Lemma 4.4 implies that $\sigma(c)\in\sigma(L(T(v_a)))$.

We claim $\sigma(d) \in \sigma(L(T(v_a)))$. We assume for contradiction that this is not the case and show that this implies the existence of an ugly quartet $\langle cdc'a' \rangle$ containing cd as its first edge, which leads to a contradiction to our initial assumption that none of the edges in $\mathcal{P}$ is the first, resp., middle edge of an ugly, resp., good quartet. To see this, note that $\sigma(a), \sigma(c) \in \sigma(L(T(v_a)))$ and Lemma 4.3 imply that there is an edge $a'c'$ for two vertices $a', c' \prec_T v_a$ with $\sigma(a') = \sigma(a)$ and $\sigma(c') = \sigma(c)$. Since $\sigma(a) = \sigma(a')$ and $\mathrm{lca}_T(a', c') \preceq_T v_a \prec_T v = \mathrm{lca}_T(a', c)$, we have $a'c \notin E(G)$. Since $\sigma(a_d) = \sigma(a')$ and $\mathrm{lca}_T(a_d, d) \preceq_T v_d \prec_T v = \mathrm{lca}_T(a', d)$, we have $a'd \notin E(G)$. Now, $\mathcal{S}^{\cap}(c, d)$ implies that $\sigma(c) \notin \sigma(L(T(v_d)))$. This and $\sigma(d) \notin \sigma(L(T(v_a)))$ together with Lemma 4.4 implies that there is an edge $c'd \in E(G)$. Thus, we obtain the ugly quartet $\langle cdc'a' \rangle$ and hence, the desired contradiction. Therefore, $\sigma(d) \in \sigma(L(T(v_a)))$. Because of $\mathcal{S}^{\cap}(a, b) = \emptyset$ we also have $\sigma(d) \notin \sigma(L(T(v_b)))$.

Since $\sigma(d) \in \sigma(L(T(v_a)))$, there is a vertex $d_a \preceq_T v_a$ with $\sigma(d_a) = \sigma(d)$. Moreover, $\sigma(b) \notin \sigma(L(T(v_a))$ and $\sigma(d) \notin \sigma(L(T(v_b)))$ together with Lemma 4.4 implies that $bd_a \in E(G)$. Furthermore, $\sigma(c) \in \sigma(L(T(v_a)))$ and Lemma 4.4 imply that $cd_a \notin E(G)$. Now, $\mathcal{S}^{\cap}(c, d) = \emptyset$ implies $\sigma(d) \notin \sigma(L(T(v_c)))$ and therefore, $\mathrm{lca}_T(c, d_a) = v \preceq_T \mathrm{lca}_T(c, d')$ for all $d' \in L[\sigma(d)]$. Hence, $(c, d_a) \in E(\vec{G})$.

In summary, $\langle dcbd_a \rangle$ is an induced P_4 in G. By [104, Lemma 32], every such induced P_4 forms either a good, bad, or ugly quartet in $(\vec{G}, \sigma)$ and, since $(c, d_a) \in E(\vec{G})$, we can conclude that $\langle dcbd_a \rangle$ is a bad quartet with first edge cd, which completes the proof. □

Lemma 6.24 is a somewhat stronger result than Thm. 8 in [104] that, moreover, emphasizes the connection between the structure of an RBMG (G, σ) and the *u-fp* edges in a corresponding BMG $(\vec{G}, \sigma)$ (cf. Prop. 6.3 and Cor. 6.3). In particular, it immediately implies

Corollary 6.7. [104, Thm. 8] *Let (G, σ) be an RBMG. Then, (G, σ) is a cograph if and only if all subgraphs induced by three colors are cographs.*

Proof. If (G, σ) is a cograph, then all its induced subgraphs are also cographs [57]. Conversely, if (G, σ) is not a cograph, then it contains at least one induced P_4. By Lemma 6.24, (G, σ) cannot contain only 4-colored P_4s and therefore the restriction to at least one combination of three colors contains a P_4 and is thus not a cograph. □

It is important to recall in this context, however, that the deletion of all *u-fp*-edges identified by quartets does not necessarily lead to a cograph (see Fig. 55(C) for an example). Hence, the quartets alone therefore cannot provide a complete algorithm for correcting an RBMG to an orthology graph. However, it would be interesting to compare the deletion of all *u-fp* edges in (G, σ) with finding a (minimal) edge-deletion set to obtain a cograph. These two problems are clearly distinct: The simplest example is the BMG $(\vec{G}, \sigma)$ in Fig. 50(A): its symmetric part G is already a cograph but $(\vec{G}, \sigma)$ contains the hug-edge xy, which must be deleted. Despite its practical use [133, 184], this observation relegates cograph editing [136, 197, 319] to the status of a heuristic approximation for the purpose of orthology detection.

6.5 SIMULATION RESULTS

We illustrate the potential impact of our mathematical results discussed in the previous sections with the help of simulated data. To this end, we focus on

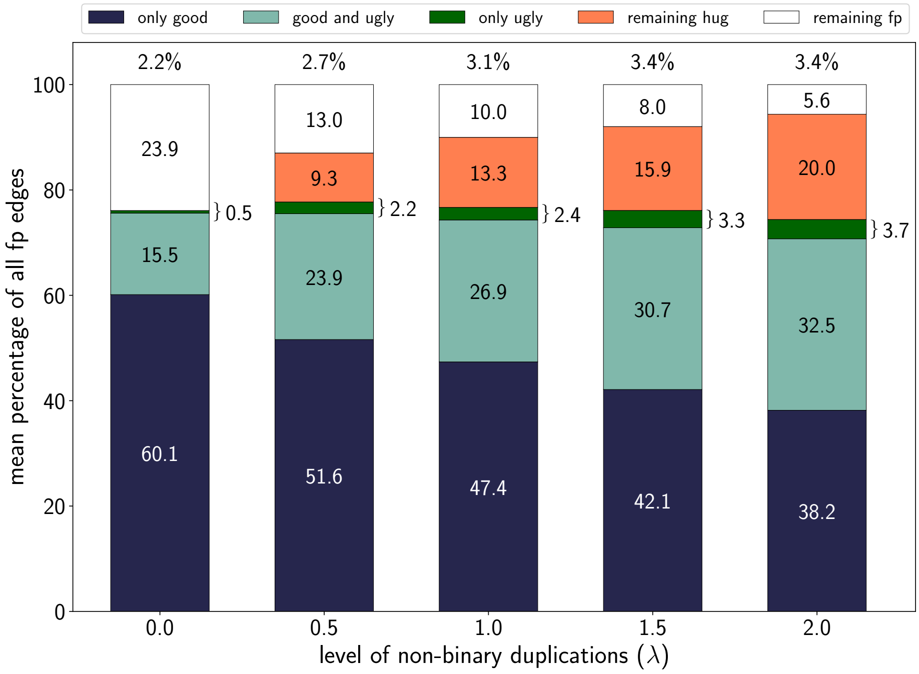


Fig. 57. Average relative abundance of the different types of hug-edges and undetectable false positives in the BMGs of simulated evolutionary scenarios. We distinguish hug-edges in good and ugly quartets as well as hug-edges appearing only in hourglass chains (orange). In the simulations, the fraction of *u-fp* edges that are first edges of bad quartets is too small too be visible and therefore not shown here. The undetectable false positives correspond to complementary gene losses without surviving witnesses of the duplication event. Species trees are binary, while gene trees contain multifurcations. The number of offsprings is modeled as $2+k$, where k is drawn from a Poisson distribution with parameter λ. For $\lambda = 0$, the gene trees are binary. In the experiments, we observed that on average 62.4% of the 25000 simulated BMGs do not contain any false-positive edge (cf. Fig. 58). Those instances are included in the computation of the fraction $|\mathfrak{F}|/|E(G)|$ (percentage above the bars). However, for the computation of all other values only scenarios that contain false-positives are considered.

the accuracy of the inferred orthology graph *assuming* that the best matches are accurate. Of course, this is only one of several components in complete orthology detection pipeline, which would also need to consider the genome annotation, pairwise alignments of genes or predicted protein sequences, and the conversion of sequence similarities into best match data. The latter step has been investigated in considerable detail by Stadler et al. [293]. Here, we start from simulated evolutionary scenarios and extract the BMG directly from the ground truth using the simulation library `AsymmeTree`, see [293] and Sec. 3.4.

We simulate pairs of dated species trees S and dated, event-labeled ("observable") gene trees (T, t, σ). In particular, we consider non-binary gene trees. This is important here since, by Lemma 4.15, hourglasses cannot appear in BMGs that are explained by a binary tree. Recall from Sec. 3.4 that the amount of non-binary duplication events in the gene tree is regulated by a

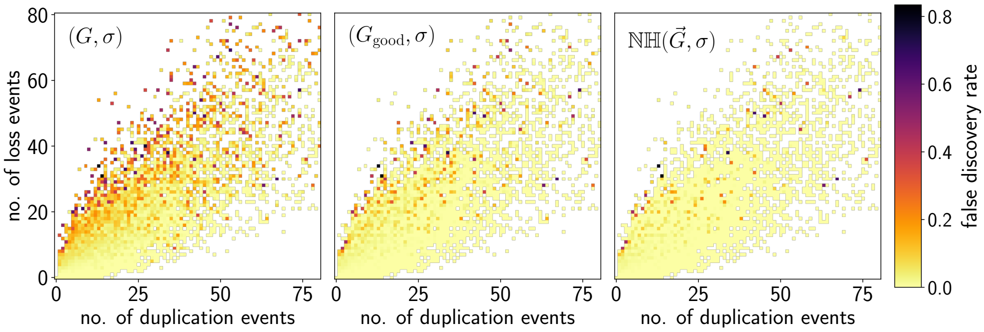


Fig. 58. False discovery rates computed as proportion of *fp* among all edges averaged over all scenarios with given number of duplications and losses. *Left:* RBMGs (G,σ), i.e., $|\mathfrak{F}|/|E(G)|$. *Middle:* edited RBMG (G_{good},σ) with all middle edges of good quartets removed, i.e., $|\mathfrak{F}\setminus\mathfrak{U}_M|/|E(G_{\text{good}})|$. *Right:* no-hug graphs $\mathbb{NH}(\vec{G},\sigma)$, i.e., $|\mathfrak{F}\setminus\mathfrak{U}|/|E(\mathbb{NH})|$. Scenarios with more than 80 duplication-loss events are not shown.

parameter $\lambda > 0$ of a Poisson distribution from which the additional number of offspring genes is drawn. The simulated data set of evolutionary scenarios comprises (binary) species trees with 10 to 30 species (drawn uniformly). The time difference between the planted root and the leaves of S is set to unity. The duplication and loss rates in the gene trees are drawn i.i.d. from the uniform distribution on the interval $[0.5, 1.5)$. Multifurcating gene trees were produced for $\lambda = \{0.0, 0.5, 1.0, 1.5, 2.0\}$. In total, we generated 5000 scenarios for each choice of λ. Since the true scenarios, and thus the true gene tree T, the true BMG $\vec{G}$, and the corresponding RBMG G are known, we can also determine the set

$$\mathfrak{F} := \{xy \mid xy \in E(G) \text{ and } t(\operatorname{lca}_T(x,y)) = \square\}. \tag{40}$$

of false-positive edges. From the BMG, we compute the set $\mathfrak{U}$ of *u-fp* edges as well as the subsets $\mathfrak{U}_M$ and $\mathfrak{U}_U$ of *u-fp* edges that are middle edges of a good or first edges of an ugly quartet, respectively. Note that in general we have $\mathfrak{U}_M \cap \mathfrak{U}_U \neq \emptyset$. We only discuss the results for binary species trees in some detail, since species trees with polytomies yield qualitatively similar results. We observe that the relative abundance of *u-fp* edges in good and ugly quartets increases moderately for larger p.

First, we note that, consistent with [103, 293], the fraction $|\mathfrak{F}|/|E(G)|$ of false positive orthology assignments is small in our data set, on the order of 3%. This indicates that, in real-life data, the main source of errors is likely the accurate determination of best matches from sequence data rather than false-positive edges contained in the BMG. Considering the fraction $|\mathfrak{U}|/|\mathfrak{F}|$ of *u-fp* edges in Fig. 57, we find that even in the most adverse case of all gene trees being binary, the BMG identifies more than three-quarters of $\mathfrak{F}$. It may be surprising at first glance that the problem becomes easier with increasing λ and barely 6% of the false positives escape discovery. A likely explanation is that multifurcations increase the likelihood that an inner vertex has two surviving lineages that serve as witnesses of the event; in addition, multifurcations increase the vertex degree in the BMG, so that in principle more information is available to resolve the tree structure. It is also interesting to note that $\mathfrak{U}_U \setminus \mathfrak{U}_M$ is small, i.e., there are few cases of first edges in an ugly quartet that are not also middle edges in a good quartet. The fraction of *u-fp* edges that appear only as first edges of bad quartets is even smaller; only 2-3% of the *u-fp* edges associated with hourglass chains, i.e., less than 0.15% of all *u-fp* edges are of this type. The overwhelming majority of *u-fp* edges associated with quartets thus appear (also) as middle edges of good quartets. This observation provides an explanation for the excellent performance of removing the $\mathfrak{U}_M$-edges proposed in [103]. In particular in the case of binary trees, which was considered by Geiß et al. [103], there is only a small number of other *u-fp* edges, which are completely covered by $\mathfrak{U}_U$. Fig. 58 visualizes the appearance of false-positive edges depending on the number of duplication and loss events. Not surprisingly, $\mathfrak{F}$ is enriched in scenarios with a large number of losses compared to the duplications, and depleted when losses are rare. In fact, in the absence of losses, the RBMG equals the orthology graph, i.e., $\mathfrak{F} = \emptyset$ [103, Thm. 4]. Removal of $\mathfrak{U}_M$, already reduced the false positives considerably.

6.6 SUMMARY

We have shown in this chapter how all unambiguously false-positive orthology assignments can be identified in polynomial time provided that all best matches are known and that we only consider HGT-free scenarios, i.e., those that can be formalized with the reconciliation map in Def. 6.1. In particular, we have provided several characterizations for *u-fp* edges in terms of underlying subgraphs and refinements of trees. Since the best match graph contains only false positives, we have obtained a characterization of *all* unambiguously incorrect orthology assignments. These results emphasize that BMGs should be considered as the first step in graph-based orthology detection methods rather than RBMGs.

Simulations showed that the majority of false positives comprises middle edges of good quartets, while *u-fp* edges that appear only as first edges of an ugly quartet are rare. Not surprisingly, the hourglass-related *u-fp* edges become important in gene trees with many multifurcations. They do not appear in scenarios derived from binary gene trees. For the theory developed here, it makes no difference whether polytomies in the gene tree appear as genuine features, or whether limited accuracy of the approximation from underlying sequence data produced the equivalent of a soft polytomy in the BMG.

The augmented tree $(\mathcal{A}(T^*), \sigma)$ of an LRT (T^*, σ) is the tree that admits an event labeling such that all inner vertices with subtrees that have overlapping colors are designated as duplications while all inner vertices with color-disjoint subtrees are designated as speciations. The tree $(\mathcal{A}(T^*), \sigma)$ therefore does not contain "non-apparent duplications" in the sense of [183], i.e., duplication vertices with species-disjoint subtrees. This is an interesting connection linking the literature concerned with polytomy refinement in given gene trees [46, 183] with best match graphs.

7

LATER-DIVERGENCE-TIME (LDT) GRAPHS

HORIZONTAL GENE TRANSFER (HGT) laterally introduces foreign genetic material into a genome. The phenomenon is particularly frequent in prokaryotes [224, 291] but also contributed to shaping eukaryotic genomes [6, 152, 167, 192, 215, 267], see also Sec. 3.2. HGT may be additive, in which case its effect is similar to gene duplications, or lead to the replacement of a vertically inherited homolog. From a phylogenetic perspective, HGT leads to an incongruence of gene trees and species trees, thus complicating the analysis of gene family histories.

In Sec. 3.3.4, both parametric and phylogenetic methods for HGT inference have been discussed briefly. The subclass of implicit (or indirect) phylogenetic methods use unexpectedly small or large distances between genes as indicators of HGT. More precisely, consider a family of homologous genes in a set of species and plot the phylogenetic distance of pairs of most similar homologs as a function of the phylogenetic distances between the species in which they reside. Since distances between orthologous genes can be expected to be approximately proportional to the distances between the species, orthologous pairs fall onto a regression line that defines equal divergence time for the last common ancestor of corresponding gene and species pairs. The gene pairs with "later divergence times", i.e., those that are more closely related than expected from their species, fall below the regression line [229]. Kanhere and Vingron [162] complemented this idea with a statistical test based on the Cook distance to identify xenologous pairs in a statistically sound manner. For the mathematical analysis, we assume that we can perfectly identify all pairs of genes a and b that are more closely related than expected from the phylogenetic distance of their respective genomes. Naturally, this defines a graph (G, σ), whose vertices x (the genes) are colored by the species $\sigma(x)$ in which they appear. Here, we are interested in two questions:

(1) What are the mathematical properties that characterize these "*later-divergence-time*" (*LDT*) graphs?

(2) What kind of information about HGT events, the gene and species tree, and the reconciliation map between them is contained implicitly in an LDT graph?

These questions are also motivated by a series of recent investigations of the mathematical structure of orthology and the (reciprocal) best match relation, see e.g. [102, 104, 132, 180] and the previous chapters, as well as the xenology relation *sensu* Fitch [101, 131, 135]. Each of these relations satisfies stringent mathematical conditions that – at least in principle – can be used to correct empirical estimates and thus serve as a potential means of noise reduction, see [133, 293] and Chapter 5.

Previous work following this paradigm, including our findings in Chapter 6, has largely been confined to duplication-loss (DL) scenarios, excluding horizontal transfer. As shown in [130], it is possible to partition a gene set into HGT-free classes separated by HGTs. Within each class, the reconstruction problems then simplify to the much easier DL scenarios. In particular, therefore, the theory developed in Chapter 6 can be applied. It is of utmost interest, therefore, to find robust methods to infer this partition directly from (dis)similarity data. In this and the next chapter, we explore the usefulness and limitations of LDT graphs for this purpose.

This chapter is organized as follows. We first introduce in Sec. 7.1 *relaxed scenarios*, a very general framework to describe evolutionary scenarios that emphasizes time consistency of reconciliation rather than particular types of evolutionary events. In Secs. 7.2–7.4, LDT graphs are defined formally and characterized as those properly colored cographs for which a set of accompanying rooted triples is consistent (Thm. 7.3). The proof is constructive and provides a method (Alg. 7) to compute a relaxed scenario for a given LDT graph. Least resolved trees for LDT graphs are studied in Sec. 7.5. In Sec. 7.6, we will briefly consider the situation that later-divergence-time information is fraught with experimental errors. The results in this chapter appeared previously in [265].

7.1 RELAXED RECONCILIATION MAPS

In Sec. 6.1, we have already introduced a reconciliation map for duplication-loss scenarios, of which we made extensive use to investigate the interdependencies of orthology and (reciprocal) best matches. However, these results are no longer valid if horizontal gene transfers are considered as possible events that have shaped the histories of gene families.

A spectrum of different mathematical frameworks have been proposed in the literature that account for horizontal transfers [24, 81, 82, 103, 108, 228]. In comparison with formalizations of duplication-loss scenarios, their design is usually more complex. To a large extent, this is because, in the presence of HGT, the relationships of gene trees T and species trees S are not only constrained by local conditions corresponding to the admissible local evolutionary events (duplication, speciation, gene loss, and HGT) but also by the global condition that the HGT events only occur between co-existing species branches and, thus, must be time-consistent in this sense, i.e., they must admit a temporal order on the species tree vertices [107, 212, 316].

Tofigh et al. [316] and Bansal et al. [24] define "Duplication-Transfer-Loss" (DTL) scenarios in terms of a vertex-only map $\gamma : V(T) \to V(S)$. The axiom set of DTL-scenarios alone does not guarantee time-consistency of a reconciliation. In particular, it was shown in [316] that finding most parsimonious DTL-scenarios is NP-hard. A definition of a DTL-like class of scenarios in terms of a reconciliation map $\mu : V(T) \to V(S) \cup E(S)$ was analyzed by Nøjgaard et al. [228]. For binary trees, the two definitions are equivalent; for non-binary trees, however, the DTL-scenarios are a proper subset, see [228, Fig. 1] for an example. Moreover, Nøjgaard et al. [228] characterize time-consistent reconciliation maps in terms of a certain auxiliary graph on the union of the vertices $V(S) \cup V(T)$ being a directed acyclic graph (DAG).

The H-trees introduced by Górecki [108], see also [110], formalize evolutionary scenarios in a very different manner. H-trees comprise binary trees B which can be seen as event-labeled gene trees with leaves denoting both surviving genes and loss events, a set of species labels $\mathcal{I}$, a set of "transfers" H, a map $\mathcal{A}$ which determines the location of the transfers in the species trees (represented by pairs of subsets of $\mathcal{I}$), and finally, a so-called dependency relation on H. The latter establishes the time-consistency constraint as a direct part of the framework. More precisely, Górecki [108] shows that this guarantees the existence of a species graph, i.e., a subdivision of a binary species tree endowed with a set of HGT edges between the single-child vertices which, loosely speaking, do not cross. Subdivisions of the species tree have e.g. also been used by Doyon et al. [81] as a means of ensuring time consistency.

Several other mathematical frameworks have been used in the literature to specify evolutionary scenarios. Examples include maps $g : V(S') \to 2^{V(T)}$ from a suitable subdivision S' of the species tree S to the gene tree as used by Hallett and Lagergren [123], and associations of edges, i.e., subsets of $E(T) \times E(S)$, as proposed by Wieseke et al. [330].

In order to capture time consistency from the outset and to establish the mathematical framework, we consider here trees with explicit timing information [212].

Definition 7.1 (Time Map). *The map $\tau_T : V(T) \to \mathbb{R}$ is a time map for a tree T if $x \prec_T y$ implies $\tau_T(x) < \tau_T(y)$ for all $x, y \in V(T)$.*

It is important to note that only *qualitative*, relative timing information will be used in practice, i.e., we will never need the actual value of time maps but only information on whether an event pre-dates, post-dates, or is concurrent with another. Def. 7.1 ensures that the ancestor relation $\preceq_T$ and the timing of the vertices are not in conflict. For later reference, we provide the following simple result.

Lemma 7.1. *Given a tree T, a time map τ_T for T satisfying $\tau_T(x) = \tau_0(x)$ with arbitrary choices of $\tau_0(x)$ for all $x \in L(T)$ can be constructed in linear time.*

Proof. We traverse T in postorder. If x is a leaf, we set $\tau_T(x) = \tau_0(x)$, and otherwise compute $t := \max_{u \in \mathsf{child}_T(x)} \tau_T(u)$ and set $\tau_T(x) = t'$ with an arbitrary value $t' > t$. Clearly the total effort is $O(|V(T)| + |E(T)|)$, and thus also linear in the number of leaves $L(T)$. □

Lemma 7.1 will be useful for the construction of time maps as it, in particular, allows us to put $\tau_T(x) = \tau_T(y)$ for all $x, y \in L(T)$.

Definition 7.2 (Time Consistency). *Let T and S be two trees. A map $\mu \colon V(T) \to V(S) \cup E(S)$ is called* time-consistent *if there are time maps τ_T for T and τ_S for S satisfying the following conditions for all $u \in V(T)$:*

(C1) If $\mu(u) \in V(S)$, then $\tau_T(u) = \tau_S(\mu(u))$.

(C2) Else, if $\mu(u) = xy \in E(S)$, then $\tau_S(y) < \tau_T(u) < \tau_S(x)$.

Conditions (C1) and (C2) ensure that the reconciliation map μ preserves time in the following sense: If vertex u of the gene tree is mapped to a vertex $\mu(u) = v$ in the species tree, then u and v receive the same time stamp by Condition (C1). If u is mapped to an edge $\mu(u) = xy$, then the time stamp of u falls within the time range $[\tau_S(x), \tau_S(y)]$ of the edge xy in the species tree. The following definition of reconciliation is designed (1) to be general enough to encompass the notions of reconciliation that have been studied in the literature, and (2) to separate the mapping between gene tree and species tree from specific types of events. Event types such as duplication or horizontal transfer therefore are considered here as a matter of *interpreting* scenarios, not as part of their definition.

Definition 7.3 (Relaxed Reconciliation Map)**.** *Let T and S be two planted trees with leaf sets $L(T)$ and $L(S)$, respectively and let $\sigma : L(T) \to L(S)$ be a map. A map $\mu\colon V(T) \to V(S) \cup E(S)$ is a* relaxed reconciliation map *for (T, S, σ) if the following conditions are satisfied:*

(G0) Root Constraint. *$\mu(x) = 0_S$ if and only if $x = 0_T$.*

(G1) Leaf Constraint. *$\mu(x) = \sigma(x)$ if and only if $x \in L(T)$.*

(G2) Time Consistency Constraint. *The map μ is time-consistent for some time maps τ_T for T and τ_S for S.*

Condition (G0) is used to map the respective planted roots. (G1) ensures that genes are mapped to the species in which they reside. (G2) enforces time consistency. The reconciliation maps most commonly used in the literature, see e.g. [24, 316], usually not only satisfy (G0)–(G2) but also impose additional conditions. We therefore call the map μ defined here "relaxed".

Definition 7.4 (Relaxed Scenario)**.** *The 6-tuple $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ is a* relaxed scenario *if μ is a relaxed reconciliation map for (T, S, σ) that satisfies (G2) w.r.t. the time maps τ_T and τ_S.*

By definition, relaxed reconciliation maps are time-consistent. Moreover, $\tau_T(x) = \tau_S(\sigma(x))$ holds for all $x \in L(T)$ by Def. 7.2(C1) and Def. 7.3(G1,G2). In the following, we will refer to the map $\sigma : L(T) \to L(S)$ as the *coloring of* $\mathcal{S}$.

7.2 LDT GRAPHS AND μ-FREE SCENARIOS

In the absence of horizontal gene transfer, the last common ancestor of two species A and B should mark the latest possible time point at which two genes a and b residing in $\sigma(a) = A$ and $\sigma(b) = B$, respectively, may have diverged. Situations in which this constraint is violated are therefore indicative of HGT. To address this issue in some more detail, we next define "μ-free scenarios" that eventually will lead us to the class of "LDT graphs" that contain all information about genes that diverged after the species in which they reside.

Definition 7.5 (μ-free scenario)**.** *Let T and S be planted trees, $\sigma\colon L(T) \to L(S)$ be a map, and τ_T and τ_S be time maps of T and S, respectively, such that $\tau_T(x) = \tau_S(\sigma(x))$ for all $x \in L(T)$. Then, $\mathcal{T} = (T, S, \sigma, \tau_T, \tau_S)$ is called a* μ-free scenario.

This definition of a scenario without a reconciliation map μ is mainly a technical convenience that simplifies the arguments in various proofs by avoiding the construction of a reconciliation map. It is motivated by the observation that the "later-divergence-time" of two genes in comparison with their species is independent from any such μ. Every relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ implies an underlying μ-free scenario $\mathcal{T} = (T, S, \sigma, \tau_T, \tau_S)$. Statements proved for μ-free scenarios therefore also hold for relaxed scenarios. Note that, by Lemma 7.1, given the time map τ_S, one can easily construct a time map τ_T such that $\tau_T(x) = \tau_S(\sigma(x))$ for all $x \in L(T)$. In particular, when constructing relaxed scenarios explicitly, we may simply choose $\tau_T(u) = 0$ and $\tau_S(x) = 0$ as common time for all leaves $u \in L(T)$ and $x \in L(S)$. Although not all μ-free scenarios admit a reconciliation map and thus can be turned into relaxed scenarios, Lemma 7.2 below implies that for every μ-free scenario $\mathcal{T}$ there is a relaxed scenario with possibly slightly distorted time maps that encodes the same LDT graph as $\mathcal{T}$.

Definition 7.6 (LDT graph)**.** *For a μ-free scenario $\mathcal{T} = (T, S, \sigma, \tau_T, \tau_S)$, we define $G_<(\mathcal{T}) = G_<(T, S, \sigma, \tau_T, \tau_S) = (V, E)$ as the graph with vertex set $V := L(T)$ and edge set*

$$E := \{ab \mid a, b \in L(T), \tau_T(\mathrm{lca}_T(a, b)) < \tau_S(\mathrm{lca}_S(\sigma(a), \sigma(b))).\}$$

A vertex-colored graph (G, σ) is a later-divergence-time graph (LDT graph)*, if there is a μ-free scenario $\mathcal{T} = (T, S, \sigma, \tau_T, \tau_S)$ such that $G = G_<(\mathcal{T})$. In this case, we say that $\mathcal{T}$* explains *(G, σ).*

It is easy to see that the edge set of $G_<(\mathcal{T})$ defines an *undirected* graph and that two genes a and b form an edge if the divergence time of a and b is strictly less than the divergence time of the underlying species $\sigma(a)$ and $\sigma(b)$. Moreover, there are no edges of the form aa, since $\tau_T(\mathrm{lca}_T(a, a)) = \tau_T(a) = \tau_S(\sigma(a)) = \tau_S(\mathrm{lca}_S(\sigma(a), \sigma(a)))$. Hence, $G_<(\mathcal{T})$ is a simple graph.

By definition, every relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ satisfies $\tau_T(x) = \tau_S(\sigma(x))$ all $x \in L(T)$. Therefore, removing μ from $\mathcal{S}$ yields a μ-free scenario $\mathcal{T} = (T, S, \sigma, \tau_T, \tau_S)$. Thus, we will use the following simplified notation.

Definition 7.7. *We put $G_<(\mathcal{S}) := G_<(T, S, \sigma, \tau_T, \tau_S)$ for a given relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ and the underlying μ-free scenario $(T, S, \sigma, \tau_T, \tau_S)$ and say, by slight abuse of notation, that $\mathcal{S}$* explains *$(G_<(\mathcal{S}), \sigma)$.*

The next two results show that the existence of a reconciliation map μ does not impose additional constraints on LDT graphs.

Lemma 7.2. *For every μ-free scenario $\mathcal{T} = (T, S, \sigma, \tau_T, \tau_S)$, there is a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \widetilde{\tau}_T, \widetilde{\tau}_S)$ for T, S and σ such that $(G_<(\mathcal{T}), \sigma) = (G_<(\mathcal{S}), \sigma)$.*

Proof. Let $\mathcal{T} = (T, S, \sigma, \tau_T, \tau_S)$ be a μ-free scenario. In order to construct a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \widetilde{\tau}_T, \widetilde{\tau}_S)$ that satisfies $G_<(\mathcal{S}) = G_<(\mathcal{T})$, we start with a time map $\widetilde{\tau}_T$ for T satisfying $\widetilde{\tau}_T(0_T) = \max(\tau_T(0_T), \tau_S(0_S))$ and $\widetilde{\tau}_T(v) = \tau_T(v)$ for all $v \in V(T) \setminus \{0_T\}$. Correspondingly, we introduce a time map $\widetilde{\tau}_S$ for S such that $\widetilde{\tau}_S(0_S) = \max(\tau_T(0_T), \tau_S(0_S))$ and $\widetilde{\tau}_S(v) = \tau_S(v)$ for all $v \in V(S) \setminus \{0_S\}$. By construction, we have $t_{\max,T} := \max\{\tau_T(v) \mid v \in V(T)\} = \tau_T(0_T) = \tau_S(0_S)$. Moreover, we have $t_{\min,S} := \min\{\tau_S(v) \mid v \in V(S)\} \leq \min\{\tau_T(v) \mid v \in V(T)\} =: t_{\min,T}$. To see this, we can choose $x \in V(T)$ such that $\tau_T(v) = t_{\min,T}$. By the definition of time maps and minimality of $\tau_T(v)$, the vertex x must be a leaf. Hence, since $\mathcal{T}$ is a μ-free scenario, we have $\tau_T(x) = \tau_S(\sigma(x))$ with $X := \sigma(x) \in L(S) \subset V(S)$. Therefore, it must hold that $t_{\min,S} \leq t_{\min,T}$. We now define $P := \{p \in V(S) \cup E(S) \mid X \preceq_S p\}$, i.e., the set of all vertices and edges on the unique path in S from 0_S to the leaf X. Since $\tau_S(X) = t_{\min,T} < t_{\max,T} = \tau_S(0_S)$, we find, for each $v \in V(T)$, *either* a vertex $u \in P$ such that $\tau_T(v) = \tau_S(u)$ *or* an edge $uw \in P$ such that $\tau_S(w) < \tau_T(v) < \tau_S(u)$. Hence, we can specify the reconciliation map μ by defining, for every $v \in V(T)$,

$$\mu(v) := \begin{cases} 0_S & \text{if } v = 0_T, \\ \sigma(v) & \text{if } v \in L(T), \\ u & \text{if there is some vertex } u \in P \text{ with } \tau_T(v) = \tau_S(u), \\ uw & \text{if there is some edge } uw \in P \text{ with } \tau_S(w) < \tau_T(v) < \tau_S(u). \end{cases}$$

For each $v \in V^0(T)$, exactly one of the two alternatives for P applies, hence μ is well-defined. It is now an easy task to verify that all conditions in Definitions 7.2 and 7.3 are satisfied for $\mathcal{S} = (T, S, \sigma, \mu, \widetilde{\tau}_T, \widetilde{\tau}_S)$ by construction. Hence, by Def. 7.4, $\mathcal{S}$ is a relaxed scenario.

It remains to show that $G_<(\mathcal{T}) = G_<(\mathcal{S})$. Let $a, b \in L(T)$ be arbitrary. Clearly, neither $\operatorname{lca}_T(a,b)$ nor $\operatorname{lca}_S(\sigma(a), \sigma(b))$ equals the planted root 0_T or 0_S, respectively. Since we have only changed the timing of the roots 0_T or 0_S, we obtain $ab \in E(G_<(\mathcal{S}))$ if and only if $\widetilde{\tau}_T(\operatorname{lca}_T(a,b)) = \tau_T(\operatorname{lca}_T(a,b)) < \widetilde{\tau}_S(\operatorname{lca}_S(\sigma(a),\sigma(b))) = \tau_S(\operatorname{lca}_S(\sigma(a),\sigma(b)))$ if and only if $ab \in E(G_<(\mathcal{T}))$, which completes the proof. ☐

Theorem 7.1. *(G,σ) is an LDT graph if and only if there is a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ such that $(G,\sigma) = (G_<(\mathcal{S}), \sigma)$.*

Proof. By definition, (G,σ) is an LDT graph for every relaxed scenario $\mathcal{S}$ with coloring σ that satisfies $(G,\sigma) = (G_<(\mathcal{S}),\sigma)$. Now suppose that (G,σ) is an LDT graph. By definition, there is a μ-free scenario $\mathcal{T} = (T, S, \sigma, \tau_T, \tau_S)$ such that $(G,\sigma) = (G_<(\mathcal{T}),\sigma)$. By Lemma 7.2, there is a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \widetilde{\tau}_T, \widetilde{\tau}_S)$ for T, S and σ such that $(G,\sigma) = (G_<(\mathcal{S}),\sigma)$. ☐

Remark 7.1. *From here on, we omit the explicit reference to Lemma 7.2 and Thm. 7.1 and assume that the reader is aware of the fact that every LDT graph is explained by some relaxed scenario $\mathcal{S}$ and that for every μ-free scenario $\mathcal{T} = (T, S, \sigma, \tau_T, \tau_S)$, there is a relaxed scenario $\mathcal{S}$ for T, S and σ such that $(G_<(\mathcal{T}),\sigma) = (G_<(\mathcal{S}),\sigma)$.*

We now derive some simple properties of μ-free and relaxed scenarios. It may be surprising at first glance that "the speciation nodes", i.e., vertices $u \in V^0(T)$ with $\mu(u) \in V(S)$ do not play a special role in determining LDT graphs.

Lemma 7.3. *For every relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ there exists a relaxed scenario $\widetilde{\mathcal{S}} = (T, S, \sigma, \widetilde{\mu}, \widetilde{\tau}_T, \tau_S)$ such that $G_<(\widetilde{\mathcal{S}}) = G_<(\mathcal{S})$ and for all distinct $x, y \in L(T)$ with $xy \notin E(G_<(\mathcal{S}))$ holds $\widetilde{\tau}_T(\operatorname{lca}_T(x,y)) > \tau_S(\operatorname{lca}_S(\sigma(x),\sigma(y)))$.*

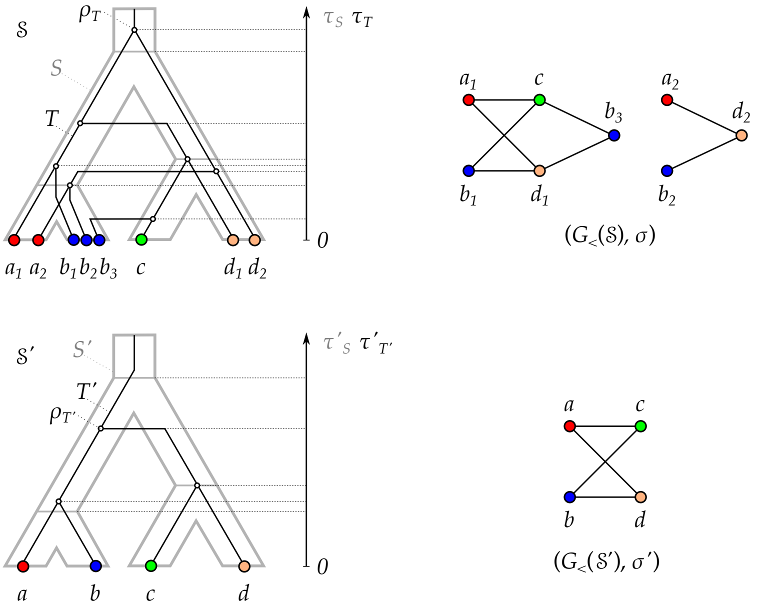


Fig. 59. Top row: A relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ (left) with its LDT graph $(G_<(\mathcal{S}), \sigma)$ (right). The reconciliation map μ is shown implicitly by the embedding of the gene tree T into the species tree S. The times τ_T and τ_S are indicated by the position on the vertical axis, i.e., if a vertex x is drawn higher than a vertex y, this implies $\tau_T(y) < \tau_T(x)$. In subsequent, figures we will not show the time maps explicitly. Bottom row: Another relaxed scenario $\mathcal{S}' = (T', S', \sigma', \mu', \tau'_T, \tau'_S)$ with a connected LDT graph $(G_<(\mathcal{S}'), \sigma')$. As we shall see, connectedness of an LDT graph depends on the relative timing of the roots of the gene and species tree (cf. Lemma 7.11).

Proof. For the relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ we write $V^0(S) := V(S) \setminus (L(S) \cup \{0_S\})$ and define

$$\begin{aligned}
D_S &:= \{|\tau_S(y) - \tau_S(x)| \colon x, y \in V(S), \tau_S(x) \neq \tau_S(y)\},\\
D_T &:= \{|\tau_T(y) - \tau_T(x)| \colon x, y \in V(T), \tau_T(x) \neq \tau_T(y)\}, \text{ and}\\
D_{TS} &:= \{|\tau_T(x) - \tau_S(y)| \colon x \in V(T),\, y \in V(S), \tau_T(x) \neq \tau_S(y)\}.
\end{aligned}$$

We have $D_S \neq \emptyset$ and $D_T \neq \emptyset$ since we do not consider empty trees, and thus, at least the "planted" edges $0_S\rho_S$ and $0_T\rho_T$ always exist. By construction, all values in D_T, D_S, and D_{TS} are strictly positive. Now define

$$\epsilon := \frac{1}{2}\min(D_{ST} \cup D_S \cup D_T).$$

Since D_S and D_T are not empty, ϵ is well-defined and, by construction, $\epsilon > 0$. Next we set, for all $v \in V(T)$,

$$\widetilde{\tau_T}(v) := \begin{cases} \tau_T(v) + \epsilon, \text{ if } v \in V^0(T) \\ \tau_T(v), \text{ otherwise,} \end{cases}$$

$$\widetilde{\mu}(v) := \begin{cases} \mathrm{par}_S(x)x, \text{ if } \mu(v) = x \in V^0(S) \\ \mu(v), \text{ otherwise.} \end{cases}$$

Claim 7.1.1. *$\widetilde{\mathcal{S}} := (T, S, \sigma, \widetilde{\mu}, \widetilde{\tau_T}, \tau_S)$ is a relaxed scenario.*

Proof of Claim: By construction, if $\mu(v) \in (L(S) \cup \{0_S\})$ and thus, $\mu(v) \notin V^0(S)$, $\mu(v)$ and $\widetilde{\mu}(v)$ coincide. Therefore, (G0) and (G1) are trivially satisfied for $\widetilde{\mu}$. In order to show (G2), we first note that $\widetilde{\tau_T}(v) = \tau_T(v) = \tau_S(\sigma(v))$ holds for all $v \in L(T)$ by Def. 7.2.

We next argue that $\widetilde{\tau_T}$ is a time map. To this end, let $x, y \in V(T)$ with $x \prec_T y$. Hence, $\tau_T(x) < \tau_T(y)$ and, in particular, $\tau_T(y) - \tau_T(x) \geq 2\epsilon$. Assume for contradiction that $\widetilde{\tau_T}(x) \geq \widetilde{\tau_T}(y)$. This implies $\widetilde{\tau_T}(x) = \tau_T(x) + \epsilon$ and $\widetilde{\tau_T}(y) = \tau_T(y)$, since $\tau_T(x) < \tau_T(y)$ and $\epsilon > 0$ always implies $\tau_T(x) + \epsilon < \tau_T(y) + \epsilon$ and $\tau_T(x) < \tau_T(y) + \epsilon$. Therefore, $\widetilde{\tau_T}(y) - \widetilde{\tau_T}(x) = \tau_T(y) - (\tau_T(x) + \epsilon) \geq \epsilon > 0$ and thus, $\widetilde{\tau_T}(y) > \widetilde{\tau_T}(x)$; a contradiction.

We continue with showing that the two time maps $\widetilde{\tau_T}$ and τ_S are time-consistent w.r.t. $\widetilde{\mathcal{S}}$. To see that Condition (C1) is satisfied, observe that, by construction, $\widetilde{\mu}(v) \in V(S)$ does hold only in case $\mu(v) \notin E(S) \cup V^0(S)$ and thus, $\mu(v) \in L(S) \cup \{0_S\}$. In this case, $\widetilde{\mu}(v) = \mu(v)$ and since $\mu(v)$ satisfies (G1) we have $v \in L(T) \cup \{0_T\}$. Thus, $v \notin V^0(T)$ and, therefore, $\widetilde{\tau_T}(v) = \tau_T(v) = \tau_S(\mu(v))$. Therefore, Condition (C1) is satisfied.

Now consider Condition (C2). As argued above, $\widetilde{\mu}(v) \in E(S)$ holds for all $v \in V^0(T) = V(T) \setminus (L(T) \cup \{0_T\})$. By construction, $\widetilde{\tau_T}(v) = \tau_T(v) + \epsilon$. There are two cases: $\mu(v) = x \in V^0(S)$, or $\mu(v) = yx \in E(S)$ with $y = \mathrm{par}_S(x)$. The following arguments hold for both cases: We have $\widetilde{\mu}(v) = yx \in E(S)$. Moreover, $\tau_S(x) \leq \tau_T(v) < \widetilde{\tau_T}(v)$ since τ_T and τ_S satisfy (C1) and (C2). Furthermore, $\tau_T(v) < \tau_S(y)$ and, by construction, $\tau_S(y) - \tau_T(v) \geq 2\epsilon$. This immediately implies that $\tau_S(y) \geq \tau_T(v) + 2\epsilon = \widetilde{\tau_T}(v) + \epsilon > \widetilde{\tau_T}(v)$. In summary, $\tau_S(x) < \widetilde{\tau_T}(v) < \tau_S(y)$ whenever $\widetilde{\mu}(v) = yx \in E(S)$. Therefore, Condition (C2) is satisfied for $\widetilde{\mathcal{S}}$. ◇

Claim 7.1.2. *$E(G_<(\mathcal{S})) \subseteq E(G_<(\widetilde{\mathcal{S}}))$.*

Proof of Claim: Let xy be an edge in $G_<(\mathcal{S})$ and thus $x \neq y$, and set $v_T := \mathrm{lca}_T(x, y)$ and $v_S := \mathrm{lca}_S(\sigma(x), \sigma(y))$. By definition, we have $\tau_T(v_T) < \tau_S(v_S)$. Therefore, we have $\tau_S(v_S) - \tau_T(v_T) \in D_{TS}$ and, hence, $\tau_S(v_S) - \tau_T(v_T) \geq 2\epsilon$. Since $x \neq y$, $v_T = \mathrm{lca}_T(x, y)$ is an inner vertex of T. By construction, therefore, $\widetilde{\tau_T}(v_T) = \tau_T(v_T) + \epsilon$. The latter arguments together with the fact that τ_S remains unchanged imply that $\tau_S(v_S) - \widetilde{\tau_T}(v_T) \geq \epsilon > 0$, and thus, $\widetilde{\tau_T}(v_T) < \tau_S(v_S)$. Therefore, we conclude that xy is an edge in $G_<(\widetilde{\mathcal{S}})$. ◇

It remains to show

Claim 7.1.3. *For all distinct $x, y \in L(T)$ with $xy \notin E(G_<(\mathcal{S}))$, we have $\widetilde{\tau_T}(\mathrm{lca}_T(x, y)) > \tau_S(\mathrm{lca}_S(\sigma(x), \sigma(y)))$.*

Proof of Claim: Suppose $xy \notin E(G_<(\mathcal{S}))$ for two distinct $x, y \in L(T)$, and set $v_T := \mathrm{lca}_T(x, y)$ and $v_S := \mathrm{lca}_S(\sigma(x), \sigma(y))$. By definition, this implies $\tau_T(v_T) \geq \tau_S(v_S)$. Since $x \neq y$, we clearly have that $v_T = \mathrm{lca}_T(x, y)$ is an inner vertex of T, and hence, $\widetilde{\tau_T}(v_T) = \tau_T(v_T) + \epsilon$. The latter two argument together with $\epsilon > 0$ and the fact that τ_S remains unchanged imply that $\widetilde{\tau_T}(v_T) > \tau_S(v_S)$. ◇

In particular, therefore, $xy \notin E(G_<(\mathcal{S}))$ implies that $xy \notin E(G_<(\widetilde{\mathcal{S}}))$ and therefore, $E(G_<(\widetilde{\mathcal{S}})) \subseteq E(G_<(\mathcal{S}))$. Together with Claim 7.1.2 and the fact that both $G_<(\mathcal{S})$ and $G_<(\widetilde{\mathcal{S}})$ have vertex set $L(T)$, we conclude that $G_<(\mathcal{S}) = G_<(\widetilde{\mathcal{S}})$, which completes the proof. ☐

Since the relaxed scenario $\widetilde{\mathcal{S}} = (T, S, \sigma, \widetilde{\mu}, \widetilde{\tau_T}, \tau_S)$ as constructed in the proof of Lemma 7.3 satisfies $\widetilde{\mu}(v) \notin V^0(S)$ we obtain

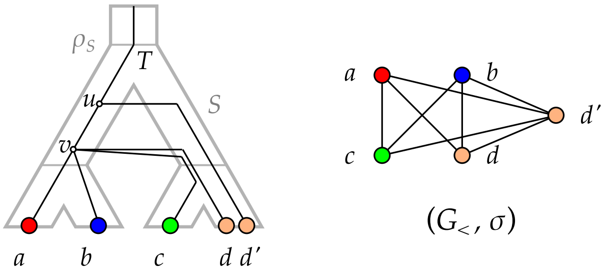


Fig. 60. Left a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ with corresponding graph $(G_<(\mathcal{S}), \sigma)$ (right). For $(G_<(\mathcal{S}), \sigma)$ there is no relaxed scenario $\widetilde{\mathcal{S}} = (T, S, \sigma, \widetilde{\mu}, \widetilde{\tau_T}, \tau_S)$ such that $G_<(\widetilde{\mathcal{S}}) = G_<(\mathcal{S})$ and for all distinct $x, y \in L(T)$ with $xy \notin E(G_<(\mathcal{S}))$ it holds that $\widetilde{\mu}(\mathrm{lca}_T(x, y)) \succ_S \mathrm{lca}_S(\sigma(x), \sigma(y))$, see Example 7.1.

Corollary 7.1. *For every relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$, there exists a relaxed scenario $\widetilde{\mathcal{S}} = (T, S, \sigma, \widetilde{\mu}, \widetilde{\tau_T}, \tau_S)$ such that $G_<(\widetilde{\mathcal{S}}) = G_<(\mathcal{S})$ and $\widetilde{\mu}(v) \notin V^0(S)$ for all $v \in V(T)$.*

Lemma 7.3, however, does not imply that one can always find a relaxed scenario with a reconciliation map $\widetilde{\mu}$ for given trees T and S satisfying $\widetilde{\mu}(\mathrm{lca}_T(x, y)) \succ_S \mathrm{lca}_S(\sigma(x), \sigma(y))$ for all distinct $x, y \in L(T)$ with $xy \notin E(G_<(\mathcal{S}))$, as shown in Example 7.1.

Example 7.1. *Consider the LDT graph $(G_<(\mathcal{S}), \sigma)$ with corresponding relaxed scenario $\mathcal{S}$ as shown in Fig. 60. Note first that $v = \mathrm{lca}_T(a, b) = \mathrm{lca}_T(c, d)$ and $ab, cd \notin E(G_<)$. To satisfy both $\widetilde{\mu}(v) \succ_S \mathrm{lca}_S(\sigma(a), \sigma(b))$ and $\widetilde{\mu}(v) \succ_S \mathrm{lca}_S(\sigma(c), \sigma(d))$, we clearly need that $\widetilde{\mu}(v) \succeq_S \rho_S$, and thus $\widetilde{\tau_T}(v) \geq \widetilde{\tau_S}(\rho_S)$. However, $ad' \in E(G_<)$ and $\mathrm{lca}_T(a, d') = u$ imply that $\widetilde{\tau_T}(u) < \tau_S(\sigma(a), \sigma(d)) = \tau_S(\rho_S)$. Hence, we obtain $\widetilde{\tau_T}(u) < \tau_S(\rho_S) \leq \widetilde{\tau_T}(v)$; a contradiction to $uv \in E(T)$ and $\widetilde{\tau_T}$ being a time map for T. Therefore, there is no relaxed scenario $\widetilde{\mathcal{S}} = (T, S, \sigma, \widetilde{\mu}, \widetilde{\tau_T}, \tau_S)$ such that $G_<(\widetilde{\mathcal{S}}) = G_<(\mathcal{S})$ and such that $\widetilde{\mu}(\mathrm{lca}_T(x, y)) \succ_S \mathrm{lca}_S(\sigma(x), \sigma(y))$ for all distinct $x, y \in L(T)$ with $xy \notin E(G_<(\mathcal{S}))$.*

For the special case that the graph under consideration has no edges we have

Lemma 7.4. *For an edgeless graph G and for any choice of T and S with $L(T) = V(G)$ and $\sigma(L(T)) = L(S)$ there is a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ that satisfies $G = G_<(\mathcal{S})$.*

Proof. Given T and S we construct a relaxed scenario as follows. Let τ_S be an arbitrary time map on S. Then we can choose τ_T such that $\tau_S(\rho_S) < \tau_T(u) < \tau_S(0_S)$ for all $u \in V^0(T)$. Each leaf $u \in L(T)$ then has a parent in T located above the last common ancestor ρ_S of all species in which case $G_<(\mathcal{S})$ is edgeless. □

Lemma 7.4 is reminiscent of the fact that for DL-only scenarios any given gene tree T can be reconciled with an arbitrary species tree as long as $\sigma(L(T)) = L(S)$ [103, 115].

7.3 PROPERTIES OF LDT GRAPHS

We continue by deriving several interesting characteristics LDT graphs.

Proposition 7.1. *Every LDT graph (G,σ) is properly colored.*

Proof. Let $\mathcal{T} = (T, S, \sigma, \tau_T, \tau_S)$ be a μ-free scenario such that $(G,\sigma) = (G_<(\mathcal{T}),\sigma)$ and recall that every μ-free scenario satisfies $\tau_T(x) = \tau_S(\sigma(x))$ for all $x \in L(T)$ with $\sigma(x) \in L(S)$. Let $a,b \in L(T)$ be distinct and suppose that $\sigma(a) = \sigma(b) = A$. Since a and b are distinct we have $a, b \prec_T \operatorname{lca}_T(a,b)$ and hence, by Def. 7.1, $\tau_T(a) < \tau_T(\operatorname{lca}_T(a,b))$. This implies that $\tau_T(a) = \tau_S(A) = \tau_S(\operatorname{lca}_S(A,A)) < \tau_T(\operatorname{lca}_T(a,b))$. Therefore, $ab \notin E(G)$. Consequently, $ab \in E(G)$ implies $\sigma(a) \neq \sigma(b)$, which completes the proof. □

Extending earlier work of Dekker (1986), Bryant and Steel (1995) derived conditions under which two triples r_1, r_2 imply a third triple r_3 that must be displayed by any tree that displays r_1, r_2. In particular, we make frequent use of the following

Lemma 7.5. *If a tree T displays $xy|z$ and $zw|y$ then T displays $xy|w$ and $zw|x$. In particular $T_{|\{x,y,z,w\}} = ((x,y),(z,w))$ (in* Newick *format).*

As we shall see below, LDT graphs (G,σ) contain detailed information about both the underlying gene trees T and species trees S for *all* μ-scenarios that explain (G,σ), and thus by Lemma 7.2 and Thm. 7.1 also about every relaxed scenario $\mathcal{S}$ satisfying $G = G_<(\mathcal{S})$. This information is encoded in the form of certain rooted triples that can be retrieved directly from local features in the colored graphs (G,σ).

Definition 7.8. *For a graph $G = (L,E)$, we define the set of triples on L as*

$$\mathfrak{T}(G) := \{xy|z : x,y,z \in L \text{ are pairwise distinct, } xy \in E,\ xz, yz \notin E\}.$$

If G is endowed with a coloring $\sigma\colon L \to M$ we also define a set of color triples

$$\mathfrak{S}(G,\sigma) := \{\sigma(x)\sigma(y)|\sigma(z) : x,y,z \in L,\ \sigma(x),\sigma(y),\sigma(z) \text{ are pairwise distinct, } xz, yz \in E,\ xy \notin E\}.$$

Lemma 7.6. *If a graph (G,σ) is an LDT graph, then $\mathfrak{S}(G,\sigma)$ is consistent and S displays $\mathfrak{S}(G,\sigma)$ for every μ-free scenario $\mathcal{T} = (T, S, \sigma, \tau_T, \tau_S)$ that explains (G,σ).*

Proof. Suppose that $(G = (L,E),\sigma)$ is an LDT graph and let $\mathcal{T} = (T, S, \sigma, \tau_T, \tau_S)$ be a μ-free scenario that explains (G,σ). In order to show that $\mathfrak{S}(G,\sigma)$ is consistent it suffices to show that S displays every triple in $\mathfrak{S}(G,\sigma)$.

Let $AB|C \in \mathfrak{S}(G,\sigma)$. By definition, A,B,C are pairwise distinct and there must be vertices $a,b,c \in L$ with $\sigma(a) = A$, $\sigma(b) = B$, and $\sigma(c) = C$ such that $ab \notin E$ and $bc, ac \in E$. First, $ab \notin E$ and $bc, ac \in E$ imply $\tau_T(\operatorname{lca}_T(a,b)) \geq \tau_S(\operatorname{lca}_S(A,B))$, $\tau_T(\operatorname{lca}_T(b,c)) < \tau_S(\operatorname{lca}_S(B,C))$, and $\tau_T(\operatorname{lca}_T(a,c)) < \tau_S(\operatorname{lca}_S(A,C))$. Moreover, for any three vertices a,b,c in T it holds that $1 \leq |\{\operatorname{lca}_T(a,b), \operatorname{lca}_T(a,c), \operatorname{lca}_T(b,c)\}| \leq 2$.

Therefore we have to consider the following four cases: (1) $u := \operatorname{lca}_T(a,b) = \operatorname{lca}_T(b,c) = \operatorname{lca}_T(a,c)$, (2) $u := \operatorname{lca}_T(a,b) = \operatorname{lca}_T(a,c) \neq \operatorname{lca}_T(b,c)$ and (3) $u := \operatorname{lca}_T(a,b) = \operatorname{lca}_T(b,c) \neq \operatorname{lca}_T(a,c)$, (4) $\operatorname{lca}_T(a,b) \neq u := \operatorname{lca}_T(b,c) = \operatorname{lca}_T(a,c)$. Note, for any three vertices x,y,z in T, $\operatorname{lca}_T(x,y) \neq \operatorname{lca}_T(x,z) = \operatorname{lca}_T(y,z)$ implies that $\operatorname{lca}_T(x,y) \prec_T \operatorname{lca}_T(x,z) = \operatorname{lca}_T(y,z)$. In Cases (1) and (2), we find $\tau_S(\operatorname{lca}_S(A,C)) > \tau_T(u) \geq \tau_S(\operatorname{lca}_S(A,B))$. Together with the fact that $\operatorname{lca}_S(A,C)$ and

$\mathrm{lca}_S(A,B)$ are comparable in S, this implies that $AB|C$ is displayed by S. In Case (3), we obtain $\tau_S(\mathrm{lca}_S(B,C)) > \tau_T(u) \geq \tau_S(\mathrm{lca}_S(A,B))$ and, by analogous arguments, $AB|C$ is displayed by S. Finally, in Case (4), the tree T displays the triple $ab|c$. Thus, $\tau_S(\mathrm{lca}_S(A,B)) \leq \tau_T(\mathrm{lca}_T(a,b)) < \tau_T(u) < \tau_S(\mathrm{lca}_S(A,C))$. Again, $AB|C$ is displayed by S. ☐

The next lemma shows that induced $K_2 + K_1$ (the disjoint union of a K_2 and a K_1) subgraphs in LDT graphs imply triples that must be displayed by the gene tree T.

Lemma 7.7. *If (G,σ) is an LDT graph, then $\mathfrak{T}(G)$ is consistent and T displays $\mathfrak{T}(G)$ for every μ-free scenario $\mathcal{T} = (T,S,\sigma,\tau_T,\tau_S)$ that explains (G,σ).*

Proof. Suppose that $(G = (L,E),\sigma)$ is an LDT graph and let $\mathcal{T} = (T,S,\sigma,\tau_T,\tau_S)$ be a μ-free scenario that explains (G,σ). In order to show that $\mathfrak{T}(G)$ is consistent it suffices to show that T displays every triple in $\mathfrak{T}(G,\sigma)$.

Let $ab|c \in \mathfrak{T}(G)$. By definition, $a,b,c \in L(T)$ are distinct, and $ab \in E$ and $ac, bc \notin E$. Since $ab \in E$, we have $A := \sigma(a) \neq \sigma(b) =: B$ by Prop. 7.1.

There are two cases, either $\sigma(c) \in \{A,B\}$ or not. Suppose first that w.l.o.g. $\sigma(c) = A$. In this case, $ab \in E$ and $bc \notin E$ together imply $\tau_T(\mathrm{lca}_T(a,b)) < \tau_S(\mathrm{lca}_S(A,B)) \leq \tau_T(\mathrm{lca}_T(b,c))$. This and the fact that $\mathrm{lca}_T(a,b)$ and $\mathrm{lca}_T(b,c)$ are comparable in T implies that T displays $ab|c$.

Suppose now that $\sigma(c) = C \notin \{A,B\}$. We now consider the four possible topologies of $S' = S_{|ABC}$: (1) S' is a star, (2) $S' = AB|C$, (3) $S' = AC|B$, and (4) $S' = BC|A$.

In Cases (1), (2) and (4), we have $\tau_S(\mathrm{lca}_S(A,B)) \leq \tau_S(\mathrm{lca}_S(A,C))$, where equality holds only in Cases (1) and (4). This together with $ab \in E$ and $ac \notin E$ implies $\tau_T(\mathrm{lca}_T(a,b)) < \tau_S(\mathrm{lca}_S(A,B)) \leq \tau_S(\mathrm{lca}_S(A,C)) \leq \tau_T(\mathrm{lca}_T(a,c))$. This and the fact that $\mathrm{lca}_T(a,b)$ and $\mathrm{lca}_T(a,c)$ are comparable in T implies that T displays $ab|c$. In Case (3), $ab \in E$ and $bc \notin E$ imply $\tau_T(\mathrm{lca}_T(a,b)) < \tau_S(\mathrm{lca}_S(A,B)) = \tau_S(\mathrm{lca}_S(B,C)) \leq \tau_T(\mathrm{lca}_T(b,c))$. By analogous arguments as before, T displays $ab|c$. ☐

We note, finally, that the Aho graph of the triple set $[\mathfrak{T}(G), L]$ in a sense recapitulates G. More precisely, we have:

Proposition 7.2. *Let $(G = (L,E),\sigma)$ be a vertex-colored graph. If for all edges $xy \in E$ there is a vertex z such that $xz, yz \notin E$ (and thus, in particular, in case that G is disconnected), then $[\mathfrak{T}(G), L] = G$.*

Proof. Clearly, the vertex sets of $[\mathfrak{T}(G), L]$ and G are the same, that is, L. Let $xy \in E$ and thus, we have $x \neq y$. There is a vertex $z \neq x,y$ in G with $xz, yz \notin E$ if and only if $xy|z \in \mathfrak{T}(G)$ and thus, if and only if xy is an edge in $[\mathfrak{T}(G), L] = G$. ☐

Definition 7.9. *For a vertex-colored graph (G,σ), we will use the shorter notation $x_1 - x_2 - \cdots - x_n$ and $X_1 - X_2 - \cdots - X_n$ for a path P_n that is induced by the vertices $\{x_i \mid 1 \leq i \leq n\}$ with colors $\sigma(x_i) = X_i$, $1 \leq i \leq n$ and edges $x_i x_{i+1}$, $1 \leq i \leq n-1$.*

The next results shows that LDT graphs cannot contain induced P_4s.

Lemma 7.8. *Every LDT graph (G,σ) is a properly colored cograph.*

Proof. Let $\mathcal{T} = (T,S,\sigma,\tau_T,\tau_S)$ be a μ-free scenario that explains (G,σ). By Prop. 7.1, (G,σ) is properly colored. To show that $G = (L,E)$ is a cograph it suffices to show that G does not contain an induced path on four vertices (cf. Prop. 2.6). Hence, assume for contradiction that G contains an induced P_4.

First we observe that for each edge ab in this P_4 it holds that $\sigma(a) \neq \sigma(b)$ since, otherwise, by Prop. 7.1, $ab \notin E$. Based on possible colorings of the P_4 w.r.t. σ and up to symmetry, we have to consider four cases: (1) $A-B-C-D$, (2) $A-B-C-A$, (3) $A-B-A-C$ and (4) $A-B-A-B$.

In Case (1) the P_4 is of the form $a-b-c-d$ with $\sigma(a) = A$, $\sigma(b) = B$, $\sigma(c) = C$, $\sigma(d) = D$. By Lemma 7.6, the species tree S must display both $AC|B$ and $BD|C$. Hence, by Lemma 7.5, $S_{|ABCD} = ((A,C),(B,D))$ in *Newick* format. Let $x := \mathrm{lca}_S(A,B,C,D) = \rho_{S_{|ABCD}}$. Note, x "separates" A and C from B and D. Now, $ab \in E$ and $ad \notin E$ implies that $\tau_T(\mathrm{lca}_T(a,b)) < \tau_S(x) \leq \tau_T(\mathrm{lca}_T(a,d))$. This and the fact that $\mathrm{lca}_T(a,b)$ and $\mathrm{lca}_T(a,d)$ are comparable in T implies that T displays $ab|d$. Similarly, $cd \in E$ and $ad \notin E$ implies that T displays $cd|a$ is displayed by T. By Lemma 7.5, $T_{|abcd} = ((a,b),(c,d))$. Let $y := \mathrm{lca}_T(a,b,c,d) = \rho_{T_{|abcd}}$. Now, $bc \in E$, $\mathrm{lca}_T(b,c) = y$, and $\mathrm{lca}_S(B,C) = x$ implies $\tau_T(y) < \tau_S(x)$. This and $\mathrm{lca}_T(a,d) = y$ and $\mathrm{lca}_S(A,D) = x$ imply that $ad \in E$, and thus a,b,c,d do not induce a P_4 in G; a contradiction.

Case (2) can be directly excluded, since Lemma 7.6 implies that, in this case, S must display $AC|B$ and $AB|C$; a contradiction.

Now consider Case (3), that is, the P_4 is of the form $a-b-a'-c$ with $\sigma(a) = \sigma(a') = A$, $\sigma(b) = B$ and $\sigma(c) = C$. By Lemma 7.6, the species tree S must display $BC|A$ and thus $x := \mathrm{lca}_S(A,B) = \mathrm{lca}_S(A,C)$. Since $ab \in E$ and $ac \notin E$ we observe $\tau_T(\mathrm{lca}_T(a,b)) < \tau_S(x) \leq \mathrm{lca}_T(a,c)$ and, as in Case (1) we infer that T displays $ab|c$. By similar arguments, $a'c \in E$ and $ac \notin E$ implies that T displays $a'c|a$. By Lemma 7.5, $T_{|abcd} = ((a,b),(a',c))$ and thus, $y := \mathrm{lca}_T(a',b) = \mathrm{lca}_T(a,c)$ and $a'b \in E$ implies that $\tau_T(y) < \tau_S(x)$. Since $y = \mathrm{lca}_T(a,c)$ and $\tau_T(y) < \tau_S(x) = \tau_S(\mathrm{lca}_S(A,C))$, we can conclude that $ac \in E$. Hence, a,b,c,d do not induce a P_4 in G; a contradiction.

In Case (4) the P_4 is of the form $a-b-a'-b'$ with $\sigma(a) = \sigma(a') = A$ and $\sigma(b) = \sigma(b') = B$. Now, $ab, a'b' \in E$ and $ab' \notin E$ imply that $\tau_T(\mathrm{lca}_T(a,b)), \tau_T(\mathrm{lca}_T(a',b')) < \tau_S(\mathrm{lca}_S(A,B)) \leq \tau_T(\mathrm{lca}_T(a,b'))$. Hence, by similar arguments as above, T must display $ab|b'$ and $a'b'|a$. By Lemma 7.5, $T_{abcd} = ((a,b),(a',b'))$ and thus, $y := \mathrm{lca}_T(a'b) = \mathrm{lca}_T(a,b')$. However, $a'b \notin E$ implies that $\tau_T(y) < \tau_S(\mathrm{lca}_S(A,B))$; a contradiction to $\tau_S(\mathrm{lca}_S(A,B)) \leq \tau_T(\mathrm{lca}_T(a,b'))$. □

The converse of Lemma 7.8 is not true is in general. To see this, consider the properly-colored cograph (G,σ) with vertex set $V(G) = \{a,a',b,b',c,c'\}$, edges $ab, bc, a'b', a'c'$ and coloring $\sigma(a) = \sigma(a') = A$, $\sigma(b) = \sigma(b') = B$, and $\sigma(c) = \sigma(c') = C$ with A,B,C being pairwise distinct. In this case, $\mathfrak{S}(G,\sigma)$ contains the triples $AC|B$ and $BC|A$. By Lemma 7.6, the tree S in every μ-free scenario $\mathcal{T} = (T,S,\sigma,\tau_T,\tau_S)$ or relaxed scenario $\mathcal{S} = (T,S,\sigma,\mu,\tau_T,\tau_S)$ explaining (G,σ) displays $AC|B$ and $BC|A$. Since no such scenario can exist, (G,σ) is not an LDT graph.

7.4 RECOGNITION AND CHARACTERIZATION OF LDT GRAPHS

In order to design an algorithm for the recognition of LDT graphs, we will consider partitions of the vertex set of a given input graph $(G = (L,E),\sigma)$. To construct suitable partitions, we start with the connected components of G. The coloring $\sigma\colon L \to M$ imposes additional constraints. We capture these with the help of binary relations that are defined in terms of partitions $\mathcal{C}$ of the color set M and employ them to further refine the partition of G.

Definition 7.10. *Let $(G = (L, E), \sigma)$ be a graph with coloring $\sigma\colon L \to M$. Let $\mathcal{C}$ be a partition of M, and $\mathcal{C}'$ be the set of connected components of G. We define the following binary relation $\mathfrak{R}(G, \sigma, \mathcal{C})$ by setting*

$$(x, y) \in \mathfrak{R}(G, \sigma, \mathcal{C}) \iff x, y \in L,\ \sigma(x), \sigma(y) \in C \text{ for some } C \in \mathcal{C}, \text{ and } x, y \in C' \text{ for some } C' \in \mathcal{C}'.$$

By construction, two vertices $x, y \in L$ are in relation $\mathfrak{R}(G, \sigma, \mathcal{C})$ whenever they are in the same connected component of G and their colors $\sigma(x), \sigma(y)$ are contained in the same set of the partition of M. The following result shows that the relation $\mathfrak{R} := \mathfrak{R}(G, \sigma, \mathcal{C})$ is an equivalence relation and every equivalence class of $\mathfrak{R}$ is contained in some connected component of G. In particular, each connected component of G is the disjoint union of $\mathfrak{R}$-classes.

Lemma 7.9. *Let $(G = (L, E), \sigma)$ be a graph with coloring $\sigma\colon L \to M$ and $\mathcal{C}$ be a partition of M. Then, $\mathfrak{R} := \mathfrak{R}(G, \sigma, \mathcal{C})$ is an equivalence relation and every equivalence class of $\mathfrak{R}$, or short $\mathfrak{R}$-class, is contained in some connected component of G. In particular, each connected component of G is the disjoint union of $\mathfrak{R}$-classes.*

Proof. It is easy to see that $\mathfrak{R}$ is reflexive and symmetric. Moreover, $xy, yz \in \mathfrak{R}$ implies that $\sigma(x), \sigma(y), \sigma(z)$ must be contained in the same set of the partition $\mathcal{C}$, and x, y, z must be contained in the same connected component of G. Therefore, $xy \in \mathfrak{R}$ and thus, $\mathfrak{R}$ is transitive. In summary, $\mathfrak{R}$ is an equivalence relation.

We continue with showing that every $\mathfrak{R}$-class K is entirely contained in some connected component of G. Clearly, there is a connected component C of G such that $C \cap K \neq \emptyset$. Assume, for contradiction, that $K \not\subseteq C$. Hence, G must be disconnected and, in particular, there is a second connected component C' of G such that $C' \cap K \neq \emptyset$. Hence, there is a pair $xy \in K$ such that $x \in C \cap K$ and $y \in C' \cap K$. But then x and y are in different connected components of G violating the definition of $\mathfrak{R}$; a contradiction. Hence, every $\mathfrak{R}$-class is entirely contained in some connected component of G. This and the fact the $\mathfrak{R}$-classes are disjoint implies that each connected component of G is the disjoint union of $\mathfrak{R}$-classes. ☐

The following partition of the leaf sets of subtrees of a tree S rooted at some vertex $u \in V(S)$ will be useful:

If u is not a leaf, then $\quad \mathcal{C}_S(u) := \{L(S(v)) \mid v \in \mathsf{child}_S(u)\}$

and, otherwise, $\quad \mathcal{C}_S(u) := \{\{u\}\}$.

One easily verifies that, in both cases, $\mathcal{C}_S(u)$ yields a valid partition of the leaf set $L(S(u))$. Recall that $\sigma_{|L',M'}\colon L' \to M'$ was defined as the "submap" of σ with $L' \subseteq L$ and $\sigma(L') \subseteq M' \subseteq M$.

Lemma 7.10. *Let $(G = (L, E), \sigma)$ be a properly colored cograph. Suppose that the triple set $\mathfrak{S}(G, \sigma)$ is consistent and let S be a tree on M that displays $\mathfrak{S}(G, \sigma)$. Moreover, let $L' \subseteq L$ and $u \in V(S)$ such that $\sigma(L') \subseteq L(S(u))$. Finally, set $\mathfrak{R} := \mathfrak{R}(G[L'], \sigma_{|L',L(S(u))}, \mathcal{C}_S(u))$.*

Then, for all distinct $\mathfrak{R}$-classes K and K', either $xy \in E$ for all $x \in K$ and $y \in K'$, or $xy \notin E$ for all $x \in K$ and $y \in K'$. In particular, for $x \in K$ and $y \in K'$, it holds that

$$xy \in E \iff K, K' \text{ are contained in the same connected component of } G[L'].$$

Proof. Let $\sigma\colon L \to M$ and put $\mathfrak{S} = \mathfrak{S}(G, \sigma)$. Since $\mathfrak{S}$ is a consistent triple set on M, there is a tree S on M that displays $\mathfrak{S}$. Moreover, the condition $\sigma(L') \subseteq L(S(u)) \subseteq M$ together with the fact that $\mathcal{C}_S(u)$ is a partition of $L(S(u))$ ensures that $\mathfrak{R}$ is well-defined.

Now suppose that K and K' are distinct $\mathfrak{R}$-classes. As a consequence of Lemma 7.9, we have exactly the two cases: either (i) K and K' are contained in the same connected component C of $G[L']$ or (ii) $K \subseteq C$ and $K' \subseteq C'$ for distinct components C and C' of $G[L']$.

Case (i). Assume, for contradiction, that there are two vertices $x \in K$ and $y \in K'$ with $xy \notin E$. Note that $C \subseteq L'$ and thus, $G[C]$ is an induced subgraph of $G[L']$. By Prop. 2.6, both induced subgraphs $G[L']$ and $G[C]$ are cographs. Now we can again apply Prop. 2.6 to conclude that $\operatorname{diam}(G[C]) \leq 2$. Hence, there is a vertex $z \in C$ such that $xz, zy \in E$. Since x and y are in distinct classes of $\mathfrak{R}$ but in the same connected component C of $G[L']$, $\sigma(x)$ and $\sigma(y)$ must lie in distinct sets of $\mathcal{C}_S(u)$. In particular, it must hold that $\sigma(x) \neq \sigma(y)$. The fact that $G[L']$ is properly colored together with $xz, yz \in E$ implies that $\sigma(z) \neq \sigma(x), \sigma(y)$. By definition and since $G[L']$ is an induced subgraph of G, we obtain that $\sigma(x)\sigma(y)|\sigma(z) \in \mathfrak{S}$. In particular, $\sigma(x)\sigma(y)|\sigma(z)$ is displayed by S. Since $\sigma(x)$ and $\sigma(y)$ lie in distinct sets of $\mathcal{C}_S(u)$, u must be an inner vertex, and we have $\sigma(x) \in L(S(v))$ and $\sigma(y) \in L(S(v'))$ for distinct $v, v' \in \mathsf{child}_S(u)$. In particular, it must hold that $\operatorname{lca}_S(\sigma(x), \sigma(y)) = u$. Moreover, $z \in C \subseteq L'$ and $\sigma(L') \subseteq L(S(u))$ imply that $\sigma(z) \in L(S(u))$. Taken together, the latter two arguments imply that S cannot display the triple $\sigma(x)\sigma(y)|\sigma(z)$; a contradiction.

Case (ii). By assumption, the $\mathfrak{R}$-classes K and K' are in distinct connected components of $G[L']$, which immediately implies $xy \notin E$ for all $x \in K$, $y \in K'$.

In summary, either $xy \in E$ for all $x \in K$ and $y \in K'$, or $xy \notin E$ for all $x \in K$ and $y \in K'$. Moreover, Case (i) establishes the *if*-direction and Case (ii) establishes, by means of contraposition, the *only-if*-direction of the final statement. ☐

Lemma 7.10 suggests a recursive strategy to construct a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ for a given properly-colored cograph (G, σ), which is illustrated in Fig. 61. The starting point is a species tree S displaying all the triples in $\mathfrak{S}(G, \sigma)$ that are required by Lemma 7.6. We show below that there are no further constraints on S and thus we may choose $S = \operatorname{Aho}(\mathfrak{S}(G, \sigma), L)$ and endow it with an arbitrary time map τ_S. Given (S, τ_S), we construct (T, τ_T) in top-down order. In order to reduce the complexity of the presentation and to make the algorithm more compact and readable, we will not distinguish the cases in which (G, σ) is connected or disconnected, nor whether a connected component is a superset of one or more $\mathfrak{R}$-classes. The tree T therefore will not be phylogenetic in general. We shall see, however, that this issue can be alleviated by simply suppressing all inner vertices with a single child.

The root u_T is placed above ρ_S to ensure that no two vertices from distinct connected components of G will be connected by an edge in $G_<(\mathcal{S})$. The vertices v_T representing the connected components C of G are each placed within an edge of S below ρ_S. W.l.o.g., the edges $\rho_S v_S$ are chosen such that

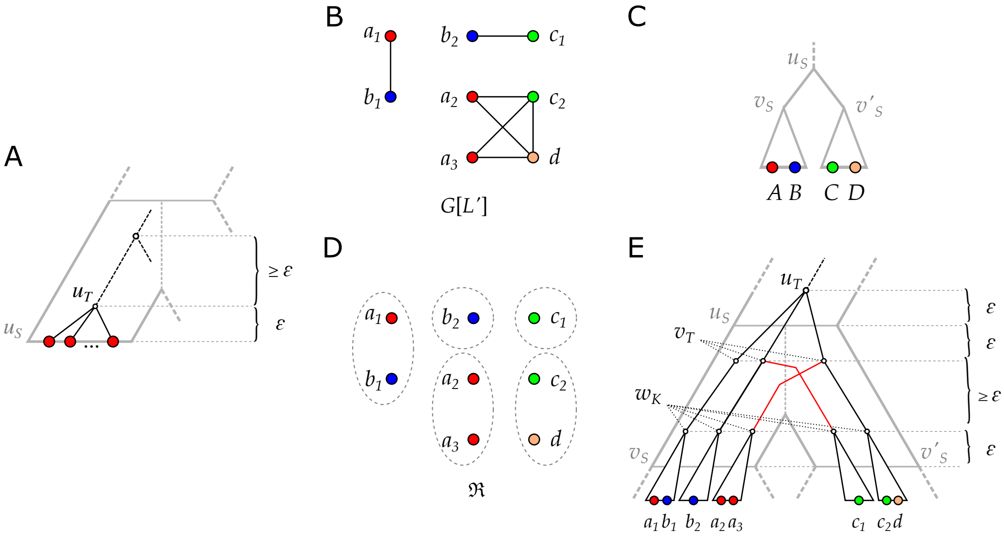


Fig. 61. Visualization of Alg. 7. (A) The case u_S is a leaf (cf. Line 8). (B)-(E) The case u_S is an inner vertex (cf. Line 12). (B) The subgraph of (G, σ) induced by L'. (C) The local topology of the species tree S yields $\mathcal{C}_S(u_S) = \{\{A, B, \dots\}, \{C, D, \dots\}\}$. Note that $L(S(u_S))$ may contain colors that are not present in $\sigma(L')$ (not shown). (D) The equivalence classes of $\mathfrak{R} := \mathfrak{R}(G[L'], \sigma_{|L', L(S(u))}, \mathcal{C}_S(u_S))$. (E) The vertex u_T and the vertices v_T are created in this recursion step. The vertices w_K corresponding to the $\mathfrak{R}$-classes K are created in the next-deeper steps. Note that some vertices have only a single child, and thus get suppressed in Line 25.

the colors of the corresponding connected component C and the colors in $L(S(v_S))$ overlap. Next we compute the relation $\mathfrak{R} := \mathfrak{R}(G, \sigma, \mathcal{C}_S(\rho_S))$ and determine, for each connected component C, the $\mathfrak{R}$-classes K that are a subset of C. For each of them, a child w_K is appended to the tree vertex v_T. The subtree $T(w_K)$ will have leaf set $L(T(w_K)) = K$. Since $\mathfrak{R}$ is defined on $\mathcal{C}_S(\rho_S)$ in this first step, $G(\mathcal{S})$ will have all edges between vertices that are in the same connected component C but in distinct $\mathfrak{R}$-classes (cf. Lemma 7.10). The definition of $\mathfrak{R}$ also implies that we always find a vertex $v_S \in \mathsf{child}_S(\rho_S)$ such that $\sigma(K) \subseteq L(S(v_S))$ (more detailed arguments for this are given in the proof of Claim 7.2.1 in the proof of Thm. 7.2 below). Thus we can place w_K into this edge $\rho_S v_S$, and proceed recursively on the $\mathfrak{R}$-classes $L' := K$, the induced subgraphs $G[L']$ and their corresponding vertices $v_S \in V(S)$, which then serve as the root of the species trees. More precisely, we identify w_K with the root u'_T created in the "next-deeper" recursion step. Since we alternate between vertices u_T for which no edges between vertices of distinct subtrees exist, and vertices v_T for which all such edges exist, we can label the vertices u_T with "0" and the vertices v_T with "1" and obtain a cotree for the cograph G.

This recursive procedure is described more formally in Alg. 7 which also describes the constructions of an appropriate time map τ_T for T and a reconciliation map μ. We note that we find it convenient to use as trivial case in the recursion the situation in which the current root u_S of the species tree is a leaf rather than the condition $|L'| = 1$. In this manner we avoid the distinction

Algorithm 7: Construction of a relaxed scenario $\mathcal{S}$ for a properly colored cograph (G,σ) with consistent triple set $\mathfrak{S}(G,\sigma)$.

```
Input: A cograph $(G = (L,E),\sigma)$ with proper coloring $\sigma\colon L \to M$ and
       consistent triple set $\mathfrak{S}(G,\sigma)$.
Output: A relaxed scenario $\mathcal{S} = (T,S,\sigma,\mu,\tau_T,\tau_S)$ explaining $(G,\sigma)$.
1  $S \leftarrow$ tree on $M$ displaying $\mathfrak{S}(G,\sigma)$ with planted root $0_S$
2  $\tau_S \leftarrow$ time map for $S$ satisfying $\tau_S(x) = 0$ for all $x \in L(S)$
3  $\epsilon \leftarrow \frac{1}{3}\min\{\tau_S(y) - \tau_S(x) \mid yx \in E(S)\}$
4  initialize empty maps $\mu, \tau_T$
5  Function BuildGeneTree($L', u_S$)
6      create a vertex $u_T$
7      $\tau_T(u_T) \leftarrow \tau_S(u_S) + \epsilon$ and $\mu(u_T) \leftarrow \mathrm{par}_S(u_S)u_S$
8      if $u_S$ is a leaf then
9          foreach $x \in L'$ do
10             connect $x$ as a child of $u_T$
11             $\tau_T(x) \leftarrow 0$ and $\mu(x) \leftarrow \sigma(x)$
12     else
13         $\mathfrak{R} \leftarrow \mathfrak{R}(G[L'], \sigma_{|L',L(S(u_S))}, \mathcal{C}_S(u_S))$
14         foreach connected component $C$ of $G[L']$ do
15             create a vertex $v_T$
16             connect $v_T$ as a child of $u_T$
17             choose $v_S^* \in \mathrm{child}_S(u_S)$ such that $\sigma(C) \cap L(S(v_S^*)) \neq \emptyset$
18             $\tau_T(v_T) \leftarrow \tau_S(u_S) - \epsilon$ and $\mu(v_T) \leftarrow u_S v_S^*$
19             foreach $\mathfrak{R}$-class $K$ such that $K \subseteq C$ do
20                 identify $v_S \in \mathrm{child}_S(u_S)$ such that $\sigma(K) \subseteq L(S(v_S))$
21                 $w_K \leftarrow$ BuildGeneTree($K, v_S$)
22                 connect $w_K$ as a child of $v_T$
23     return $u_T$
24 $T' \leftarrow$ tree with root BuildGeneTree($L, \rho_S$)
25 $T \leftarrow T'$ with (i) a planted root $0_T$ added, and (ii) all inner degree-2 vertices
     (except $0_T$) suppressed
26 $\tau_T(0_T) \leftarrow \tau_S(0_S)$ and $\mu(0_T) \leftarrow 0_S$
27 return $(T, S, \sigma, \mu_{|V(T)}, \tau_{T|V(T)}, \tau_S)$
```

between the cases $u_S \in L(S)$ and $u_S \notin L(S)$ in the **else**-condition starting in Line 12. This results in a shorter presentation at the expense of more inner vertices that need to be suppressed at the end in order to obtain the final tree T. We proceed by proving the correctness of Alg. 7.

Theorem 7.2. *Let (G,σ) be a properly colored cograph, and assume that the triple set $\mathfrak{S}(M,G)$ is consistent. Then Alg. 7 returns a relaxed scenario $\mathcal{S} = (T,S,\sigma,\mu,\tau_T,\tau_S)$ such that $G_<(\mathcal{S}) = G$ in polynomial time.*

Proof. Let $\sigma\colon L \to M$ and put $\mathfrak{S} := \mathfrak{S}(G,\sigma)$. By a slight abuse of notation, we will simply write μ and τ_T also for restrictions to subsets of $V(T)$. Observe first that due to Line 7, the algorithm continues only if (G,σ) is a properly colored cograph and $\mathfrak{S}$ is consistent, and returns a tuple $\mathcal{S} = (T,S,\sigma,\mu,\tau_T,\tau_S)$ in this case. In particular, a tree S on M that displays $\mathfrak{S}$ exists, and can e.g. be constructed using `BUILD` (Line 1). By Lemma 7.1, we can always construct a time map τ_S for S satisfying $\tau_S(x) = 0$

for all $x \in L(S)$ (Line 2). By definition, $\tau_S(y) > \tau_S(x)$ must hold for every edge $yx \in E(S)$, and thus, we obtain $\epsilon > 0$ in Line 3. Moreover, the recursive function `BuildGeneTree` maintains the following invariant:

Claim 7.2.1. *In every recursion step of the function* `BuildGeneTree`*, we have* $\sigma(L') \subseteq L(S(u_S))$.

Proof of Claim: Since S (with root ρ_S) is a tree on M by construction and thus $L(S(\rho_S)) = M$, the statement holds for the top-level recursion step on L and ρ_S. Now assume that the statement holds for an arbitrary step on L' and u_S. If u_S is a leaf, there are no deeper recursion steps. Thus assume that u_S is an inner vertex. Recall that $\mathcal{C}_S(u_S)$ is a partition of $L(S(u_S))$ (by construction), and that $\mathfrak{R} = \mathfrak{R}(G[L'], \sigma_{|L',L(S(u))}, \mathcal{C}_S(u_S))$ is an equivalence relation (by Lemma 7.9). This together with the definition of $\mathfrak{R}$ and $\sigma(L') \subseteq L(S(u_S))$, implies that there is a child $v_S \in \mathsf{child}_S(u_S)$ such that $\sigma(K) \subseteq L(S(v_S))$ for all $\mathfrak{R}$-classes K. In particular, therefore, the statement is true for all recursive calls on K and v_S in Line 21. Repeating this argument top-down along the recursion hierarchy proves the claim. ◇

Note, that we are in the *else*-condition in Line 13 only if u_S is not a leaf. Therefore and as a consequence of Claim 7.2.1 and by similar arguments as in its proof, there is a vertex $v_S^* \in \mathsf{child}_S(u_S)$ such that $\sigma(C) \cap L(S(v_S^*)) \neq \emptyset$ for every connected component C of $G[L']$ in Line 17, and a vertex $v_S \in \mathsf{child}_S(u_S)$ such that $\sigma(K) \subseteq L(S(v_S))$ for every $\mathfrak{R}$-class K in Line 20. Moreover, $\mathsf{par}_S(u_S)$ is always defined since we have $u_S = \rho_S$ and thus $\mathsf{par}_S(u_S) = 0_S$ in the top-level recursion step, and recursively call the function `BuildGeneTree` on vertices v_S such that $v_S \prec_S u_S$.

In summary, all assignments are well-defined in every recursion step. It is easy to verify that the algorithm terminates since, in each recursion step, we either have that u_S is a leaf, or we recurse on vertices v_S that lie strictly below u_S. We argue that the resulting tree T' is a *not necessarily phylogenetic* tree on L by observing that, in each step, each $x \in L'$ is either attached to the tree as a leaf if u_S is a leaf, or, since $\mathfrak{R}$ forms a partition of L' by Lemma 7.9, passed down to a recursion step on K for some $\mathfrak{R}$-class K. Nevertheless, T' is turned into a phylogenetic tree T by suppression of degree-two vertices in Line 25. Finally, $\mu(x)$ and $\tau_T(x)$ are assigned for all vertices $x \in L(T') = L$ in Line 11, and for all newly created inner vertices in Lines 7 and 18.

Recall that τ_S is a valid time map satisfying $\tau_S(x) = 0$ for all $x \in L(S)$ by construction. Before we continue to show that $\mathcal{S}$ is a relaxed scenario, we first show that the conditions for time maps and time consistency are satisfied for $(T', \tau_T, S, \tau_S, \mu)$:

Claim 7.2.2. *For all $x, y \in V(T')$ with $x \prec_{T'} y$, we have $\tau_T(x) < \tau_T(y)$. Moreover, for all $x \in V(T')$, the following statements are true:*

(i) if $\mu(x) \in V(S)$, then $\tau_T(x) = \tau_S(\mu(x))$, and

(ii) if $\mu(x) = ab \in E(S)$, then $\tau_S(b) < \tau_T(x) < \tau_S(a)$.

Proof of Claim: Recall that we always write an edge uv of a tree T such that $v \prec_T u$. For the first part of the statement, it suffices to show that $\tau_T(x) < \tau_T(y)$ holds for every edge $yx \in E(T')$, and thus to consider all vertices $x \neq \rho_{T'}$ in T' and their unique parent, which will be denoted by y in the following. Likewise, we have to consider all vertices $x \in V(T')$ including the root to show the second statement. The root $\rho_{T'}$ of T' corresponds to the vertex u_T created in Line 6 in the top-level recursion step on L and ρ_S. Hence, we have $\mu(\rho_{T'}) = \mathsf{par}_S(\rho_S)\rho_S = 0_S\rho_S \in E(S)$ and $\tau_T(\rho_{T'}) = \tau_S(\rho_S) + \epsilon$ (cf. Line 7). Therefore, we have to show (ii). Since $\epsilon > 0$, it holds that $\tau_S(\rho_S) < \tau_T(\rho_{T'})$. Moreover, $\tau_S(0_S) - \tau_S(\rho_S) \geq 3\epsilon$ holds by construction, and thus $\tau_S(0_S) - (\tau_T(\rho_{T'}) - \epsilon) \geq 3\epsilon$ and $\tau_S(0_S) - \tau_T(\rho_{T'}) \geq 2\epsilon$, which together with $\epsilon > 0$ implies $\tau_T(\rho_{T'}) < \tau_S(0_S)$.

We now consider the remaining vertices $x \in V(T') \setminus \{\rho_{T'}\}$. Every such vertex x is introduced into T' in some recursion step on L' and u_S in one of the Lines 6, 10, 15 or 21. There are exactly the following three cases: (a) $x \in L(T')$ is a leaf attached to some inner vertex u_T in Line 10, (b) $x = v_T$ as created in Line 15, and (c) $x = w_T$ as assigned in Line 21. Note that if $x = u_T$ as created in Line 6, then u_T is either the root of T', or equals a vertex w_T as assigned in Line 21 in the "parental" recursion step.

In Case (a), we have that $x \in L(T')$ is a leaf and attached to some inner vertex $y = u_T$. Since u_S must be a leaf in this case, and thus $\tau_S(u_S) = 0$, we have $\tau_T(y) = 0 + \epsilon = \epsilon$ and $\tau_T(x) = 0$ (cf. Lines 7 and 11). Since $\epsilon > 0$, this implies $\tau_T(x) < \tau_T(y)$. Moreover, we have $\mu(x) = \sigma(x) \in L(S) \subset V(S)$ (cf. Line 11), and thus have to show Subcase (i). Since u_S is a leaf and $\sigma(L') \subseteq L(S(u_S))$, we conclude $\sigma(x) = u_S$. Thus we obtain $\tau_T(x) = 0 = \tau_S(u_S) = \tau_S(\mu(x))$.

In Case (b), we have $x = v_T$ as created in Line 15, and x is attached as a child to some vertex $y = u_T$ created in the same recursion step. Thus, we have $\tau_T(y) = \tau_S(u_S) + \epsilon$ and $\tau_T(x) = \tau_S(u_S) - \epsilon$ (cf. Lines 7 and 18). Therefore and since $\epsilon > 0$, it holds $\tau_T(x) < \tau_T(y)$. Moreover, we have $\mu(x) = u_S v_S^* \in E(S)$ for some $v_S^* \in \mathsf{child}_S(u_S)$. Hence, we have to show Subcase (ii). By a similar calculation as before, $\epsilon > 0$, $\tau_S(u_S) - \tau_S(v_S^*) \geq 3\epsilon$ and $\tau_T(x) = \tau_S(u_S) - \epsilon$ imply $\tau_S(v_S^*) < \tau_T(x) < \tau_S(u_S)$.

In Case (c), $x = w_T$ as assigned in Line 21 is equal to u_T as created in Line 6 in some next-deeper recursion step with $u'_S \in \mathsf{child}_S(u_S)$. Thus, we have $\tau_T(x) = \tau_S(u'_S) + \epsilon$ and $\mu(x) = u_S u'_S \in E(S)$ (cf. Line 7). Moreover, x is attached as a child of some vertex $y = v_T$ as created in Line 15. Thus, we have $\tau_T(y) = \tau_S(u_S) - \epsilon$. By construction and since $u_S u'_S \in E(S)$, we have $\tau_S(u_S) - \tau_S(u'_S) \geq 3\epsilon$. Therefore, $(\tau_T(y) + \epsilon) - (\tau_T(x) - \epsilon) \geq 3\epsilon$ and thus $\tau_T(y) - \tau_T(x) \geq \epsilon$. This together with $\epsilon > 0$ implies $\tau_T(x) < \tau_T(y)$. Moreover, since $\mu(x) = u_S u'_S \in E(S)$ for some $u'_S \in \mathsf{child}_S(u_S)$, we have to show Subcase (ii). By a similar calculation as before, $\epsilon > 0$, $\tau_S(u_S) - \tau_S(u'_S) \geq 3\epsilon$ and $\tau_T(x) = \tau_S(u'_S) + \epsilon$ imply $\tau_S(u'_S) < \tau_T(x) < \tau_S(u_S)$.
◇

Claim 7.2.3. *$\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ is a relaxed scenario.*

Proof of Claim: The tree T is obtained from T' by first adding a planted root 0_T (and connecting it to the original root) and then suppressing all inner vertices except 0_T that have only a single child in Line 25. In particular, T is a planted phylogenetic tree by construction. The root constraint (G0) $\mu(x) = 0_S$ if and only if $x = 0_T$ also holds by construction (cf. Line 26). Since we clearly have not contracted any outer edges yx, i.e. with $x \in L(T')$, we conclude that $L(T') = L(T) = L$. As argued before, we have $\tau_T(x) = 0$ and $\mu(x) = \sigma(x)$ whenever $x \in L(T') = L(T)$ (cf. Line 11). Since all other vertices are either 0_T or mapped by μ to some edge of S (cf. Lines 26, 7 and 18), the leaf constraint (G1) $\mu(x) = \sigma(x)$ is satisfied if and only if $x \in L(T)$.

By construction, we have $V(T) \setminus \{0_T\} \subseteq V(T')$. Moreover, suppression of vertices clearly preserves the $\preceq_T$-relation between all vertices $x, y \in V(T) \setminus \{0_T\}$. Together with Claim 7.2.2, this implies $\tau_T(x) < \tau_T(y)$ for all vertices $x, y \in V(T) \setminus \{0_T\}$ with $x \prec_T y$. For the single child ρ_T of 0_T in T, we have $\tau_T(\rho_T) \leq \tau_S(\rho_S) + \epsilon$ where equality holds if the root of T' was not suppressed and thus is equal to ρ_T. Moreover, $\tau_T(0_T) = \tau_S(0_S)$ and $\tau_S(0_S) - \tau_S(\rho_S) \geq 3\epsilon$ hold by construction. Taken together the latter two arguments imply that $\tau_T(\rho_T) < \tau_T(0_T)$. In particular, we obtain $\tau_T(x) < \tau_T(y)$ for all vertices $x, y \in V(T)$ with $x \prec_T y$. Hence, τ_T is a time map for T, which, moreover, satisfies $\tau_T(x) = 0$ for all $x \in L(T)$.

To show that $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ is a relaxed scenario, it remains to show that μ is time-consistent with the time maps τ_T and τ_S. In case $x \in L(T) \subset V(T)$, we have $\mu(x) = \sigma(x) \in L(S) \subset V(S)$ and thus $\tau_T(x) = 0 = \tau_S(\sigma(x)) = \tau_S(\mu(x))$. For

0_T, we have $\tau_T(0_T) = \tau_S(0_S) = \tau_S(\mu(0_T))$. The latter two arguments imply that all vertices $x \in L(T) \cup \{0_T\}$ satisfy (C1) in the Def. 7.2. The remaining vertices of T are all vertices of T' as well. In particular, they are all inner vertices that are mapped to some edge of S (cf. Lines 7 and 18). The latter two arguments together with Claim 7.2.2 imply that, for all vertices $x \in V(T) \setminus (L(T) \cup \{0_T\})$, we have $\mu(x) = ab \in E(S)$ and $\tau_S(b) < \tau_T(x) < \tau_S(a)$. Therefore, every such vertex satisfies (C2) in Def. 7.2. It follows that the time consistency constraint (G2) is also satisfied, and thus $\mathcal{S}$ is a relaxed scenario. ◇

Claim 7.2.4. *Every vertex $v \in V^0(T)$ was either created in Line 6 or in Line 15. In particular, it holds for all $x, y \in L(T)$ with $\mathrm{lca}_T(x, y) = v$:*

(1) If v was created in Line 6, then $xy \notin E(G)$ and $xy \notin E(G_<(\mathcal{S}))$.

(2) If v was created in Line 15, then $xy \in E(G)$ and $xy \in E(G_<(\mathcal{S}))$.

Furthermore, G is a cograph with cotree (T, t) where $t(v) = 0$ if v was created in Line 6 and $t(v) = 1$, otherwise.

Proof of Claim: Since T is phylogenetic, every vertex $v \in V^0(T)$ is the last common ancestor of two leaves $x, y \in L := L(T)$. Let $v \in V^0(T)$ be arbitrary and choose arbitrary leaves $x, y \in L$ such that $\mathrm{lca}_T(x, y) = v$. Since $v \in V^0(T)$, the leaves x and y must be distinct.

Note that $v \notin L(T) \cup \{0_T\}$, and thus, v is also an inner vertex in T'. Therefore, we have exactly the two cases (1) $v = u_T$ is created in Line 6, and (2) $v = v_T$ is created in Line 15. Similar as before, the case that $v = w_K$ is assigned in Line 21 is covered by Case (a), since, in this case, w_K is created in a deeper recursion step.

We consider the recursion step on L' and u_S, in which v was created. Clearly, it must hold that $x, y \in L'$. Before we continue, set $\mathfrak{R} := \mathfrak{R}(G[L'], \sigma_{|L', L(S(u))}, \mathcal{C}_S(u_S))$ as in Line 13. Note, since $\mathcal{S}$ is a relaxed scenario, the graph $(G_<(\mathcal{S}), \sigma)$ is well-defined.

For Statement (1), suppose that $v = u_T$ was created in Line 6. Hence, we have the two cases (i) the vertex u_S of S in this recursion step is a leaf, and (ii) u_S is an inner vertex. In Case (i), we have $L(S(u_S)) = \{u_S\}$. Together with Claim 7.2.1 and $\sigma(x), \sigma(y) \in \sigma(L')$, this implies $\sigma(x) = \sigma(x) = u_S$. By assumption, (G, σ) is properly colored. By Prop. 7.1 $(G_<(\mathcal{S}), \sigma)$ must be properly colored as well. Hence, we conclude that $xy \notin E(G)$ and $xy \notin E(G_<(\mathcal{S}))$, respectively. In Case (ii), u_S is not a leaf. Therefore, $\mathrm{lca}_T(x, y) = v = u_T$ is only possible if x and y lie in distinct connected components of $G[L']$. This immediately implies $xy \notin E(G)$. Moreover, we have $\sigma(x), \sigma(y) \in L(S(u_S))$ and thus $\mathrm{lca}_S(\sigma(x), \sigma(y)) \preceq_S u_S$. Since τ_S is a time map for S, it follows that $\tau_S(\mathrm{lca}_S(\sigma(x), \sigma(y))) \leq \tau_S(u_S)$. Together with $\tau_T(u_T) = \tau_S(u_S) + \epsilon$ (cf. Line 7) and $\epsilon > 0$, this implies $\tau_S(\mathrm{lca}_S(\sigma(x), \sigma(y))) < \tau_T(v) = \tau_T(\mathrm{lca}_T(x, y))$. Hence, $xy \notin E(G_<(\mathcal{S}))$.

For Statement (2), suppose that $v = v_T$ was created in Line 15. Therefore, $\mathrm{lca}_T(x, y) = v = v_T$ is only possible if x and y lie in the same connected components of $G[L']$ but in distinct $\mathfrak{R}$-classes. Now, we can apply Lemma 7.10 to conclude that $xy \in E(G)$. Moreover, the fact that x and y lie in the same connected component of $G[L']$ but in distinct $\mathfrak{R}$-classes implies that $\sigma(x)$ and $\sigma(y)$ lie in distinct sets of $\mathcal{C}_S(u_S)$. Hence, there are distinct $v_S, v'_S \in \mathrm{child}_S(u)$ such that $\sigma(x) \preceq_S v_S$ and $\sigma(y) \preceq_S v'_S$. In particular, $\mathrm{lca}_S(\sigma(x), \sigma(y)) = u_S$. In Line 18, we assign $\tau_T(\mathrm{lca}_T(x, y)) = \tau_T(v_T) = \tau_S(u_S) - \epsilon$. Together with $\epsilon > 0$, the latter two arguments imply $\tau_T(\mathrm{lca}_T(x, y)) < \tau_S(u_S) = \tau_S(\mathrm{lca}_S(\sigma(x), \sigma(y)))$. Therefore, we have $xy \in E(G_<(\mathcal{S}))$.

By the latter arguments, the cotree (T, t) as defined above is well-defined and, for all $v \in V^0(T)$, we have $t(v) = 1$ if and only if $xy \in E(G)$ for all $x, y \in L$ with $\mathrm{lca}_T(x, y) = v$. Hence, (T, t) is a cotree for G. ◇

Claim 7.2.5. *The relaxed scenario $\mathcal{S}$ satisfies $G_<(\mathcal{S}) = G$.*

Proof of Claim: Since $L(T) = L$, the two undirected graphs $G_<(\mathcal{S})$ and G have the same vertex set. By Claim 7.2.4, we have, for all distinct $x, y \in L$, either $xy \notin E(G)$ and $xy \notin E(G_<(\mathcal{S}))$, or $xy \in E(G)$ and $xy \in E(G_<(\mathcal{S}))$. $\diamond$

Together, Claims 7.2.3 and 7.2.5 imply that Alg. 7 returns a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ such that $G_<(\mathcal{S}) = G$.

To see that Alg. 7 runs in polynomial time, we first note that the function `BuildGeneTree()` operates in polynomial time. This is clear for the setup and the **if** part. The construction of $\mathfrak{R}$ in the **else** part involves the computation of connected components and the evaluation of Def. 7.10, both of which can be achieved in polynomial time. This is also true for the comparisons of color classes required to identify v_S^* and v_S. Since the sets K in recursive calls of `BuildGeneTree()` form a partition of L', and the v_S are children of u_S in S and the depth of the recursion is bounded by $O(|L(S)|)$, the total effort remains polynomial. ☐

As a consequence of Lemma 7.6 and 7.8, and the fact that Alg. 7 returns a relaxed scenario $\mathcal{S}$ for a given properly colored cograph with consistent triple set $\mathfrak{S}(G, \sigma)$, we obtain

Theorem 7.3. *A graph (G, σ) is an LDT graph if and only if it is a properly colored cograph and $\mathfrak{S}(G, \sigma)$ is consistent.*

Proof. By Lemma 7.6 and 7.8, if (G, σ) is an LDT graph, then it is a properly colored cograph and $\mathfrak{S}(G, \sigma)$ is consistent. Now suppose that (G, σ) is a properly colored cograph and $\mathfrak{S}(G, \sigma)$ is consistent. Then, by Thm. 7.2, Alg. 7 outputs a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ such that $G_<(\mathcal{S}) = G$. By definition, this in particular implies that (G, σ) is an LDT graph. ☐

Thm. 7.3 has two consequences that are of immediate interest:

Corollary 7.2. *LDT graphs can be recognized in polynomial time.*

Proof. Cographs can be recognized in linear time [58], the proper coloring can be verified in linear time, the triple set $\mathfrak{S}(G, \sigma)$ contains not more than $|V(G)| \cdot |E(G)|$ triples and can be constructed in $O(|V(G)| \cdot |E(G)|)$ time, and consistency of $\mathfrak{S}(G, \sigma)$ can be checked in $O(\min(|\mathfrak{S}| \log^2 |V(G)|, |\mathfrak{S}| + |V(G)|^2 \ln |V(G)|))$ time [158]. ☐

Corollary 7.3. *The property of being an LDT graph is hereditary, that is, if (G, σ) is an LDT graph then each of its vertex induced subgraphs is an LDT graph.*

Proof. Let $(G = (V, E), \sigma)$ be an LDT graph. It suffices to show that $(G - x, \sigma_{|V \setminus \{x\}})$ is an LDT graph, where $G - x$ is obtained from G by removing $x \in V$ and all its incident edges. By Prop. 2.6, $G - x$ is a cograph that clearly remains properly colored. Moreover, every induced path on three vertices in $G - x$ is also an induced path on three vertices in G. This implies that if $xy|z \in \mathfrak{S}' = \mathfrak{S}(G - x, \sigma_{|V \setminus \{x\}})$, then $xy|z \in \mathfrak{S}(G, \sigma)$. Hence, $\mathfrak{S}' \subseteq \mathfrak{S}(G, \sigma)$. By Thm. 7.3, $\mathfrak{S}(G, \sigma)$ is consistent. Hence, any tree that displays all triples in $\mathfrak{S}(G, \sigma)$, in particular, displays all triples in $\mathfrak{S}'$. Therefore, $\mathfrak{S}'$ is consistent. In summary, $(G - x, \sigma_{|V \setminus \{x\}})$ is a properly colored cograph and $\mathfrak{S}'$ is consistent. By Thm. 7.3 it is an LDT graph. ☐

The relaxed scenarios $\mathcal{S}$ explaining an LDT graph (G, σ) are far from being unique. In fact, we can choose from a large set of trees (S, τ_S) that is determined only by the triple set $\mathfrak{S}(G, \sigma)$:

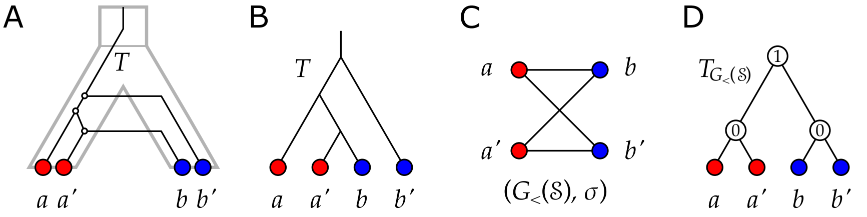


Fig. 62. A relaxed scenario $\mathcal{S}$ (A) with gene tree T (B) and its associated graph $(G_<(\mathcal{S}), \sigma)$ (C). The discriminating cotree $T_{G_<(\mathcal{S})}$ (D) is not displayed by T.

Corollary 7.4. *If $(G = (L, E), \sigma)$ is an LDT graph with coloring $\sigma : L \to M$, then for all planted trees S on M that display $\mathfrak{S}(G, \sigma)$ there is a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ that contains σ and S and that explains (G, σ).*

Proof. If (G, σ) is an LDT graph, then the species tree S assigned in Line 1 in Alg. 7 is an arbitrary tree on M displaying $\mathfrak{S}(G, \sigma)$. ☐

As shown by the next result, for every LDT graph (G, σ), there is a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ explaining (G, σ) such that T displays the discriminating cotree T_G of G.

Corollary 7.5. *If (G, σ) is an LDT graph, then there exists a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ explaining (G, σ) such that T displays the discriminating cotree T_G of G.*

Proof. Suppose that (G, σ) is an LDT graph. By Thm. 7.3, (G, σ) must be a properly colored cograph and $\mathfrak{S}(G, \sigma)$ is comparable. Hence, Thm. 7.2 implies that Alg. 7 constructs a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ explaining (G, σ). In particular, the tree T together with labeling t as specified in Claim 7.2.4 is a cotree for G. Since the unique discriminating cotree $(T_G, \hat{t})$ of G is obtained from any other cotree by contraction of edges in T, the tree T must display T_G. ☐

Although, Cor. 7.5 implies that there is always a relaxed scenario $\mathcal{S}$ where the tree T displays the discriminating cotree T_G of $G = G(\mathcal{S})$, this is not true for all relaxed scenarios $\mathcal{S}$ with $G = G(\mathcal{S})$. Fig. 62 shows a relaxed scenario $\mathcal{S}' = (T', S', \sigma, \mu', \tau'_T, \tau'_S)$ with $G = G(\mathcal{S}')$ for which T' does not display T_G.

Cor. 7.5 enables us to relate connectedness of LDT graphs to properties of the relaxed scenarios by which it can be explained.

Lemma 7.11. *An LDT graph $(G = (L, E), \sigma)$ with $|L| > 1$ is connected if and only if for every relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ that explains (G, σ), we have $\tau_T(\rho_T) < \tau_S(\mathrm{lca}_S(\sigma(L)))$.*

Proof. By contraposition, suppose first that there is a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ that explains (G, σ) such that $\tau_T(\rho_T) \geq \tau_S(\mathrm{lca}_S(\sigma(L)))$. Since $|L(T)| = |L| > 1$, the root ρ_T is not a leaf. To show that G is disconnected we consider two distinct children $v, w \in \mathrm{child}_T(\rho_T)$ of the root and leaves $x \in L(T(v))$ and $y \in L(T(w))$ and verify that x and y cannot be adjacent in G. If $\sigma(x) = \sigma(y)$, then $xy \notin E$ since (G, σ) is properly colored (cf. Lemma 7.8). Hence, suppose that $\sigma(x) \neq \sigma(y)$. By construction, $\mathrm{lca}_T(x, y) = \rho_T$ and thus, by assumption, $\tau_T(\mathrm{lca}_T(x, y)) = \tau_T(\rho_T) \geq \tau_S(\mathrm{lca}_S(\sigma(L)))$. Now $\mathrm{lca}_S(\sigma(L)) \succeq_S \mathrm{lca}_S(\sigma(x), \sigma(y))$ implies that $\tau_S(\mathrm{lca}_S(\sigma(L))) \geq \tau_S(\mathrm{lca}_S(\sigma(x), \sigma(y)))$ and thus, $\tau_T(\mathrm{lca}_T(x, y)) \geq \tau_S(\mathrm{lca}_S(\sigma(x), \sigma(y)))$. Hence, $xy \notin E$.

Consequently, for all distinct children $v, w \in \mathsf{child}_T(\rho_T)$, none of the vertices in $L(T(v))$ are adjacent to any of the vertices in $L(T(w))$ and thus, G is disconnected.

Conversely, suppose that G is disconnected. We consider Alg. 7 with input (G, σ). By Thms. 7.2 and 7.3, the algorithm constructs a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ that explains (G, σ). Consider the top-level recursion step on L and ρ_S. Since G is disconnected, the vertex u_T created in Line 6 of this step equals the root ρ_T of the final tree T. To see this, assume first that ρ_S is a leaf. Then, we attach the $|L| > 1$ elements in L as leaves to u_T (cf. Line 10). Now assume that ρ_S is not a leaf. Since $G[L] = G$ has at least two components, we attach at least two vertices v_T created in Line 15 to u_T. Hence u_T is not suppressed in Line 25 and thus $\rho_T = u_T$. By construction, therefore, we have $\tau_T(\rho_T) = \tau_T(u_T) = \tau_S(u_S) + \epsilon = \tau_S(\rho_S) + \epsilon$ for some $\epsilon > 0$. From $\sigma(\rho_S) \succeq_S \mathrm{lca}_S(\sigma(L))$ and the definition of time maps, we obtain $\tau_S(\rho_S) \geq \tau_S(\mathrm{lca}_S(\sigma(L)))$. Therefore, we have $\tau_T(\rho_T) \geq \tau_S(\mathrm{lca}_S(\sigma(L))) + \epsilon > \tau_S(\mathrm{lca}_S(\sigma(L)))$, which completes the proof. Therefore, we have shown so-far that if all relaxed scenarios $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ that explain (G, σ) satisfy $\tau_T(\rho_T) \leq \tau_S(\mathrm{lca}_S(\sigma(L)))$, then (G, σ) must be connected. However, $\tau_T(\rho_T) = \tau_S(\mathrm{lca}_S(\sigma(L)))$ cannot occur, since we can reuse the same arguments as in the beginning of this proof to show that, in this case, G is disconnected. ☐

7.5 LEAST RESOLVED TREES FOR LDT GRAPHS

As we have seen e.g. in Cor. 7.4, there are in general many trees S and T forming relaxed scenarios $\mathcal{S}$ that explain a given LDT graph (G, σ). This begs the question to what extent these trees are determined by "representatives". For S, we have seen that S always displays $\mathfrak{S}(G, \sigma)$, suggesting to consider the role of $S = \mathrm{Aho}(\mathfrak{S}(G, \sigma), M)$, where M is the codomain of σ. This tree is least resolved in the sense that there is no relaxed scenario explaining the LDT graph (G, σ) with a tree S' that is obtained from S by edge-contractions. The latter is due to the fact that any edge contraction in $\mathrm{Aho}(\mathfrak{S}(G, \sigma), M)$ yields a tree S' that does not display $\mathfrak{S}(G, \sigma)$ any more [159]. By Prop. 7.6, none of the relaxed scenarios containing S' explain the LDT graph (G, σ).

Definition 7.11. *Let $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ be a relaxed scenario explaining the LDT graph (G, σ). The planted tree T is* least resolved *for (G, σ) if no relaxed scenario $(T', S', \sigma', \mu', \tau'_T, \tau'_S)$ with $T' < T$ explains (G, σ).*

In other words, T is least resolved for (G, σ) if no relaxed scenario with a gene tree T' obtained from T by a series of edge contractions explains (G, σ).

The examples in Fig. 63 show that LDT graphs are in general not accompanied by unique least resolved trees. In the top row, relaxed scenarios with different least resolved gene trees T and the same least resolved species tree S explain the LDT graph (G, σ). In the example below, two distinct least resolved species trees exist for a given least-resolved gene tree.

The example in Fig. 64 shows, furthermore, that the unique discriminating cotree T_G of an LDT graph (G, σ) is not always "sufficiently resolved". To see this, assume that the graph (G, σ) in the example can be explained by a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ such that $T = T_G$. First consider the connected component consisting of a, b, c, d. Since $\mathrm{lca}_T(a, b) \succ_T \mathrm{lca}_T(c, d)$, $ab \in E(G)$ and $cd \notin E(G)$, we have $\tau_S(\mathrm{lca}_S(\sigma(a), \sigma(b))) > \tau_T(\mathrm{lca}_T(a, b)) > \tau_T(\mathrm{lca}_T(c, d)) \geq \tau_S(\mathrm{lca}_S(\sigma(c), \sigma(d)))$. By similar arguments, the second con-

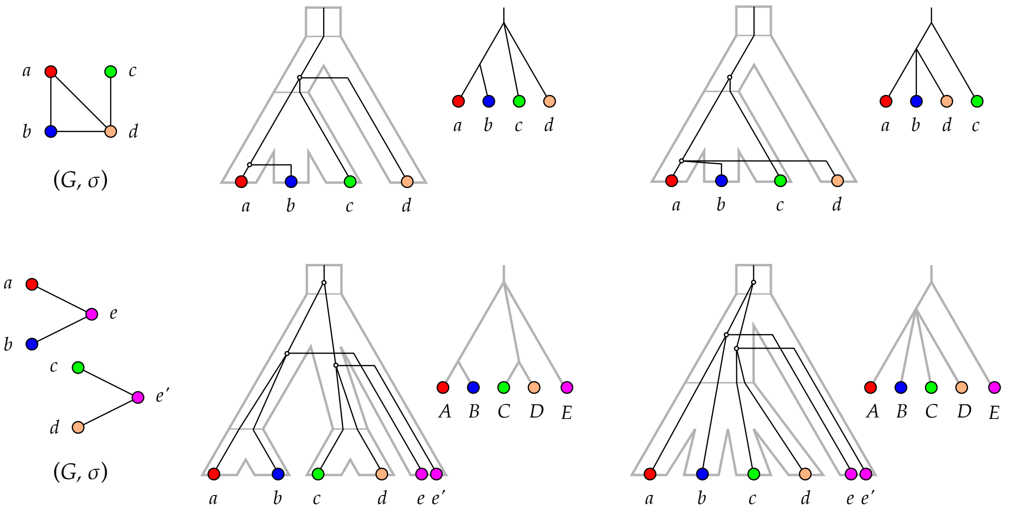


Fig. 63. Examples of LDT graphs (G,σ) with multiple least resolved trees. Top row: No unique least resolved gene tree. For both trees, contraction of the single inner edge leads to a loss of the gene triple $ab|c \in \mathfrak{T}(G)$ (cf. Lemma 7.7). The species tree is also least resolved since contraction of its single inner edge leads to loss of the species triples $\sigma(a)\sigma(c)|\sigma(d), \sigma(b)\sigma(c)|\sigma(d) \in \mathfrak{S}(G,\sigma)$ (cf. Lemma 7.6). Bottom row: No unique least resolved species tree. Both trees display the two necessary triples $AB|E, CD|E \in \mathfrak{S}(G,\sigma)$, and are again least resolved w.r.t. these triples. The gene trees are also least resolved since contraction of either of its two inner edges leads e.g. to loss of one of the triples $ae|c, ce'|a \in \mathfrak{T}(G)$.

nected component implies $\tau_S(\mathrm{lca}_S(\sigma(c),\sigma(d))) > \tau_S(\mathrm{lca}_S(\sigma(a),\sigma(b)))$; a contradiction. These examples emphasize that LDT graphs constrain the relaxed scenarios, but are far from determining them.

7.6 LDT GRAPH EDITING

Putative LDT graphs (G,σ) can be estimated directly from sequence (dis)similarity data. The most direct approach was introduced by Novichkov et al. [229], where, for (reciprocally) most similar genes x and y from two distinct species $\sigma(x) = A$ and $\sigma(x) = B$, dissimilarities $\delta(x,y)$ between genes and dissimilarities $\Delta(A,B)$ of the underlying species are compared under the assumption of a (gene family specific) clock-rate r, i.e., the expectation that orthologous gene pairs satisfy $\delta(x,y) \approx r\Delta(A,B)$. In this setting, $xy \in E(G)$ if $\delta(x,y) < r\Delta(A,B)$ at some level of statistical significance. The rate assumption can be relaxed to consider rank-order statistics. For fixed x, differences in the orders of $\delta(x,y)$ and $\Delta(\sigma(x),\sigma(y))$ assessed by rank-order correlation measures have been used to identify x as HGT candidate e.g. [55, 186]. An interesting variation on the theme is described by Sevillya et al. [274], who use relative synteny rather than sequence similarity for the same purpose. A more detailed account on estimating (G,σ) will be given elsewhere.

Empirical estimates of LDT graphs from sequence data are expected to suffer from noise and hence to violate the conditions of Thm. 7.3. It is of interest, therefore, to consider the problem of correcting an empirical estimate

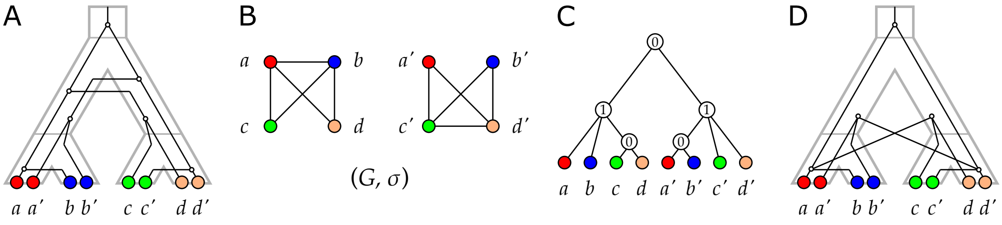


Fig. 64. Example of an LDT graph (G, σ) in Panel B that is explained by the relaxed scenario shown in Panel A. Here, (G, σ) cannot be explained by a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ such that T is the unique discriminating cotree (shown in panel C) for the cograph G, see Panel D and the text for further explanations.

(G, σ) to the closest LDT graph. We therefore briefly investigate the usual three edge *modification* problems for graphs: *completion* only considers the insertion of edges, for *deletion* edges may only be removed, while solutions to the *editing* problem allow both insertions and deletions, see e.g. [39].

Problem 7.1 (LDT-Graph-Modification (LDT-M)).

Input: *A colored graph* $(G = (V, E), \sigma)$ *and an integer* k.

Question: *Is there a subset* $F \subseteq \binom{V}{2}$ *such that* $|F| \leq k$ *and* $(G \odot F, \sigma)$ *is an LDT graph where* $\odot \in \{\triangle, +, -\}$*?*

We write LDT-E, LDT-C, LDT-D for the editing, completion, and deletion version of LDT-M. By virtue of Thm. 7.3, the LDT-M is closely related to the problem of finding a consistent subset $\mathcal{R} \subseteq \mathfrak{S}(G_{\mathcal{R}}, \sigma)$ with maximum cardinality. The corresponding decision problem, MaxRTC, is known to be NP-complete [157, Thm. 1].

NP-completeness of LDT-M be shown by reduction from

Problem 7.2 (Maximum Rooted Triple Consistency (MaxRTC)).

Input: *A set of (rooted) triples* $\mathcal{R}$ *and an integer* k.

Question: *Is there a consistent subset* $\mathcal{R}^* \subseteq \mathcal{R}$ *such that* $|\mathcal{R}^*| \geq |\mathcal{R}| - k$*?*

Theorem 7.4. [157, Thm. 1] MaxRTC *is NP-complete.*

Theorem 7.5. LDT-M *is NP-complete.*

Proof. Since LDT graphs can be recognized in polynomial time (cf. Cor. 7.2), a given solution can be verified in polynomial time. Thus, LDT-M is contained in NP.

We now show NP-hardness by reduction from MaxRTC. Let $(\mathcal{R}, k)$ be an instance of this problem, i.e., $\mathcal{R}$ is a set of triples and k is a non-negative integer. We construct a colored graph $(G_{\mathcal{R}} = (L, E), \sigma)$ as follows: For each triple $r_i = xy|z \in \mathcal{R}$, we add three vertices x_i, y_i, z_i, two edges x_iz_i and y_iz_i, and put $\sigma(x_i) = x$, $\sigma(y_i) = y$ and $\sigma(z_i) = z$. Hence, $(G_{\mathcal{R}}, \sigma)$ is properly colored and the disjoint union of paths on three vertices P_3. In particular, therefore, $(G_{\mathcal{R}}, \sigma)$ does not contain an induced P_4, and is therefore a properly colored cograph (cf. Prop. 2.6). By definition and construction, we have $\mathcal{R} = \mathfrak{S}(G_{\mathcal{R}}, \sigma)$.

First assume that MaxRTC with input $(\mathcal{R}, k)$ has a yes-answer. In this case let $\mathcal{R}^* \subseteq \mathcal{R}$ be a consistent subset such that $|\mathcal{R}^*| \geq |\mathcal{R}| - k$. For each of the triples $r_i = xy|z \in \mathcal{R} \setminus \mathcal{R}^*$, we add the edge x_iy_i to $G_{\mathcal{R}}$ or remove the edge x_iz_i from $G_{\mathcal{R}}$ for LDT-E/C and LDT-D, respectively, to obtain the graph G^*. In both cases, we eliminate the corresponding triple $xy|z$ from $\mathfrak{S}(G^*,\sigma)$. By construction, therefore, we observe that $\mathfrak{S}(G^*,\sigma) = \mathcal{R}^*$ is consistent. Moreover, since we have never added edges between distinct P_3s, all connected components of G^* are of size at most three. Therefore, G^* does not contain an induced P_4, and thus remains a cograph. By Thm. 7.3, the latter arguments imply that (G^*,σ) is an LDT graph. Since (G^*,σ) was obtained from $(G_{\mathcal{R}},\sigma)$ by using $|\mathcal{R} \setminus \mathcal{R}^*| \leq k$ edge modifications, we conclude that LDT-M with input $(G_{\mathcal{R}},\sigma,k)$ has a yes-answer.

For the converse, suppose that LDT-M with input $(G_{\mathcal{R}},\sigma,k)$ has a yes-answer with a solution $(G^* := G \odot F,\sigma)$, i.e., (G^*,σ) is an LDT graph and $|F| \leq k$. By Thm. 7.3, $\mathfrak{S}(G^*,\sigma)$ is consistent. Let $\mathcal{R}^*$ be the subset of $\mathcal{R} = \mathfrak{S}(G_{\mathcal{R}},\sigma)$ containing all triples of $\mathcal{R}$ for which the corresponding induced P_3 in $G_{\mathcal{R}}$ remains unmodified and thus, is still an induced P_3 in G^*. By construction, we have $\mathcal{R}^* \subseteq \mathfrak{S}(G^*,\sigma)$. Hence, $\mathcal{R}^*$ is consistent. Moreover, since $|F| \leq k$, at most k of the vertex-disjoint P_3s have been modified. Therefore, we conclude that $|\mathcal{R}^*| \geq |\mathcal{R}| - k$.

In summary, LDT-M is NP-hard. $\square$

We note that the NP-hardness of the LDT-M problem can also be established by a reduction from Cograph Editing/Completion/Deletion, all of which are NP-complete as well [85, 197]. Consider an instance (G,k) of one of the cograph modification problems and color *all* vertices in G differently to obtain an instance (G,σ,k) of the corresponding LDT-M problem. NP-hardness follows directly from the observation that $G \odot F$ is a cograph if and only if $(G \odot F,\sigma)$ is an LDT graph for any $F \subseteq \binom{V(G)}{2}$ and $\odot \in \{\triangle,+,-\}$. To see this, note that $(G \odot F,\sigma)$ is trivially properly colored, and consider a cotree (T,t) for a cograph $G \odot F$. We can replace every leaf $l \in L(T) = V(G)$ by its color $\sigma(l)$. The resulting cotree (T',t) is well-defined since every color appears exactly once. Now consider a triple $\sigma(a)\sigma(b)|\sigma(c) \in \mathfrak{S}(G \odot F,\sigma)$, i.e., $a - c - b$ is an induced P_3 in $G \odot F$. Full enumeration of all possible triples on $\{a,b,c\}$ and the possibilities for the t-labelings of their two inner vertices shows that T must display the triple $ab|c$. By construction, therefore, T' displays the triple $\sigma(a)\sigma(b)|\sigma(c)$. Since this is true for any triple in $\mathfrak{S}(G \odot F,\sigma)$, the latter set must be consistent. By Thm. 7.3, $(G \odot F,\sigma)$ is an LDT graph. Conversely, $G \odot F$ is a cograph for every LDT graph $(G \odot F,\sigma)$. Hence, the "hardness" of the LDT-M problem lies in both conditions of the LDT graph characterization: the cograph property and consistency of the species triples.

7.7 SUMMARY

In this chapter, we have introduced later-divergence-time (LDT) graphs as a model capturing the subset of horizontal transfer detectable through the pairs of genes that have diverged later than their respective species. Within the setting of relaxed scenarios, LDT graphs (G,σ) are exactly the properly colored cographs with a consistent triple set $\mathfrak{S}(G,\sigma)$ (Thm. 7.3). A scenario that explains a given LDT graph can be constructed in polynomial time using Alg. 7. We have seen that LDT graphs are not associated with unique least resolved gene or species trees as part of their explaining scenarios (Fig. 63). Moreover, all three modification problems for LDT graphs are NP-complete (Thm. 7.5).

8

FROM LDT GRAPHS TO XENOLOGY

LATER-DIVERGENCE-TIME GRAPHS and their mathematical properties in the framework of relaxed scenarios have been studied in the previous chapter. They contain the information about which pairs of genes diverged more recently than the respective species (or genomes) in which they reside. However, for practical applications, it would be of more valuable interest to infer whether the path connecting two genes in the gene tree contains a horizontal gene transfer, i.e., whether they are xenologs. Therefore, we now turn to the following question: How much information about HGT events and the xenology relation is contained in LDT graphs?

To this end, Sec. 8.1 formally defines HGT events and "rs-Fitch graphs", i.e., vertex-colored graph representations of xenology relations that are feasible in the relaxed scenario framework. In Sec. 8.2, we show that every edge in a LDT graph corresponds to a pair of xenologs (Thm. 8.1), and characterize those LDT graphs that already capture all HGT events. In addition, we provide a characterization of rs-Fitch graphs in terms of their coloring in Sec. 8.3. These properties can be verified in polynomial time. In order to provide an avenue for understanding rs-Fitch graphs and their explaining scenarios, we study least resolved trees of Fitch graphs (being a superclass with a somewhat simpler structure) in Sec. 8.4. Since LDT graphs do not usually capture all HGT events, we discuss in Sec. 8.5 several ways to obtain a plausible set of HGT candidates from LDT graphs. In Sec. 8.6, we finally address the question above with the help of simulations of evolutionary scenarios with a wide range of duplication, loss, and HGT event rates. Like the previous one, this chapter is also based on [265].

8.1 HGT-LABELED TREES AND RS-FITCH GRAPHS

As alluded to in the last chapter, LDT graphs are intimately related with horizontal gene transfer. To formalize this connection, we first define transfer edges. These will then be used to encode Walter Fitch's concept of xenologous gene pairs [61, 91] as a binary relation, and thus, the edge set of a graph.

Definition 8.1. *Let* $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ *be a relaxed scenario. An edge* uv *in* T *is a* transfer edge *if* $\mu(u)$ *and* $\mu(v)$ *are incomparable in* S*. The* HGT-labeling *of* T *in* $\mathcal{S}$ *is the edge labeling* $\lambda_{\mathcal{S}} : E(T) \to \{0, 1\}$ *with* $\lambda(e) = 1$ *if and only if* e *is a transfer edge.*

The vertex u in T thus corresponds to an HGT event, with v denoting the subsequent event, which now takes place in the "recipient" branch of the species tree. Note that $\lambda_{\mathcal{S}}$ is completely determined by $\mathcal{S}$. In general, for a given a gene tree T, HGT events correspond to a labeling or coloring of the edges of T.

Definition 8.2 (Fitch graph)**.** *Let* (T,λ) *be a tree* T *together with a map* $\lambda\colon E(T) \to \{0,1\}$*. The* Fitch graph $\digamma(T,\lambda) = (V,E)$ *has vertex set* $V := L(T)$ *and edge set*

$$E := \{xy \mid x,y \in L, \textit{ the unique path connecting } x \textit{ and } y \textit{ in } T \\ \textit{contains an edge } e \textit{ with } \lambda(e) = 1.\}$$

By definition, Fitch graphs of 0/1-edge-labeled trees are loopless and undirected. We call edges e of (T,λ) with label $\lambda(e) = 1$ also 1-edges and, otherwise, 0-edges.

Remark 8.1. *Fitch graphs as defined here have been termed* undirected *Fitch graphs [135], in contrast to the notion of the* directed *Fitch graphs of 0/1-edge-labeled trees studied e.g. in [101, 131].*

Proposition 8.1. [135, 344] *The following statements are equivalent.*

1. *G is the Fitch graph of a 0/1-edge-labeled tree.*
2. *G is a complete multipartite graph.*
3. *G does not contain a* $K_2 + K_1$ *as an induced subgraph.*

A natural connection between LDT graphs and complete multipartite graphs is suggested by the definition of triple sets $\mathfrak{T}(G)$, since each forbidden induced subgraph $K_2 + K_1$ of a complete multipartite graphs corresponds to a triple in an LDT graph. More precisely, we have:

Lemma 8.1. (G,σ) *is a properly colored complete multipartite if and only if it is properly colored and* $\mathfrak{T}(G) = \emptyset$*.*

Proof. The equivalence between the statements can be seen by observing that G is a complete multipartite graph if and only if G does not contain an induced $K_2 + K_1$ (cf. Prop. 8.1). By definition of $\mathfrak{T}(G)$, this is the case if and only if $\mathfrak{T}(G) = \emptyset$. ☐

Definition 8.3 (rs-Fitch graph)**.** *Let* $\mathcal{S} = (T,S,\sigma,\mu,\tau_T,\tau_S)$ *be a relaxed scenario with HGT-labeling* $\lambda_{\mathcal{S}}$*. We call the vertex colored graph* $(\digamma(\mathcal{S}),\sigma) := (\digamma(T,\lambda_{\mathcal{S}}),\sigma)$ *the* Fitch graph of the relaxed scenario $\mathcal{S}$.
A vertex colored graph (G,σ) *is a* relaxed scenario Fitch graph (rs-Fitch graph) *if there is a relaxed scenario* $\mathcal{S} = (T,S,\sigma,\mu,\tau_T,\tau_S)$ *such that* $G = \digamma(\mathcal{S})$*.*

Fig. 65 shows that rs-Fitch graphs are not necessarily properly colored. A subtle difficulty arises from the fact that Fitch graphs of 0/1-edge-labeled trees are defined without a reference to the vertex coloring σ, while the rs-Fitch graph is vertex-colored. This together with Prop. 8.1 implies

Observation 8.1. *If* (G,σ) *is an rs-Fitch graph, then* G *is a complete multipartite graph.*

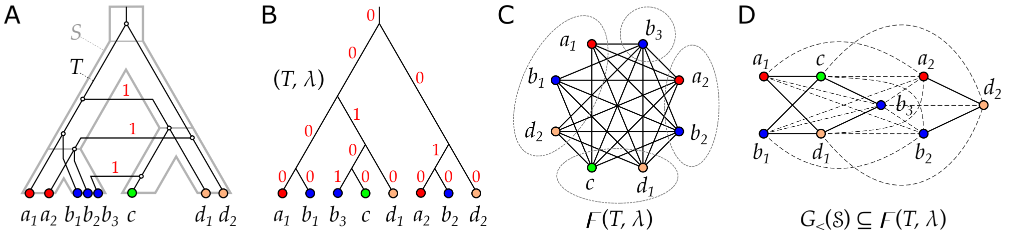

A
S
T
1
1
1
$a_1 a_2$ $b_1 b_2 b_3$ c d_1 d_2
B
(T, λ)
0
0
0
1
0
0
1
0 0 1 0 0 0 0 0
a_1 b_1 b_3 c d_1 a_2 b_2 d_2
C
a_1 b_3 b_1 a_2 d_2 b_2 c d_1
$\digamma(T, \lambda)$
D
a_1 c a_2 d_2 b_3 b_1 d_1 b_2
$G_<(\mathcal{S}) \subseteq \digamma(T, \lambda)$

Fig. 65. (A) The relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ as already shown in Fig. 59. (B) A 0/1-edge-labeled tree (T, λ) satisfying $\lambda = \lambda_{\mathcal{S}}$. (C) The corresponding Fitch graph $\digamma(T, \lambda)$ drawn in a layout that emphasizes the property that $\digamma(T, \lambda)$ is a complete multipartite graph. Independent sets are circled. (D) An alternative layout as in Fig. 59 (top row) that emphasizes the relationship $G_<(\mathcal{S}) \subseteq \digamma(\mathcal{S}) = \digamma(T, \lambda)$ (cf. Thm. 8.1 below). Edges that are not present in $G_<(\mathcal{S})$ are drawn as dashed lines.

The "converse" of Obs. 8.1 is not true in general, as we shall see in Thm. 8.3 below. If, however, the coloring σ can be chosen arbitrarily, then every complete multipartite graph G can be turned into an rs-Fitch graph (G, σ) as shown in Prop. 8.2.

Proposition 8.2. *If G is a complete multipartite graph, then there exists a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ such that (G, σ) is an rs-Fitch graph.*

Proof. Let G be a complete multipartite graph and set $L := V(G)$ and $R := E(G)$. If $R = \emptyset$, then the relaxed scenario $\mathcal{S}$ constructed in the proof of Lemma 7.4 shows that $E(G) = E(\digamma(\mathcal{S})) = \emptyset$. Hence, we assume that $R \neq \emptyset$ and explicitly construct a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ such that (G, σ) is an rs-Fitch graph.

We start by specifying the coloring $\sigma \colon L \to M$. Since G is a complete multipartite graph it is determined by its independent sets $I_1, \dots, I_k$, which form a partition of L. We set $M := \{1, 2, \dots, k\}$ and color every $x \in I_j$ with color $\sigma(x) = j$, $1 \leq j \leq k$. By construction, (G, σ) is properly colored, and $\sigma(x) = \sigma(y)$ whenever $xy \notin R$, i.e., whenever x and y lie in the same independent set. Therefore, we have $\mathfrak{S}(G, \sigma) = \emptyset$. Let S be the planted star tree with leaf set $L(S) = \{1, \dots, k\} = M$ and $\mathsf{child}_S(\rho_S) = M$. Since $R \neq \emptyset$, we have $k \geq 2$, and thus, ρ_S has at least two children and is, therefore, phylogenetic. We choose the time map τ_S by putting $\tau_S(0_S) = 2$, $\tau_S(\rho_S) = 1$ and $\tau_S(x) = 0$ for all $x \in L(S)$.

Finally, we construct the planted phylogenetic tree T with planted root 0_T and root ρ_T as follows: Vertex ρ_T has k children $u_1, \dots, u_k$. If $I_j = \{x_j\}$ consists of a single element, then we put $u_j := x_j$ as a leaf or T, and otherwise, vertex u_j has exactly $|I_j|$ children where $\mathsf{child}_T(u_j) = I_j$. Now label, for all $i \in \{2, \dots, k\}$, the edge $\rho_T u_i$ with "1", and all other edges with "0". Since $k \geq 2$, the tree T is also phylogenetic by construction.

We specify the time map τ_T and the reconciliation map μ by defining, for every $v \in V(T)$,

$$\tau_T(v) := \begin{cases} 2 = \tau_S(0_S) \\ 0 \\ 1/2 \\ 1/4 \end{cases} \qquad \mu(v) := \begin{cases} 0_S & \text{if } v = 0_T, \\ \sigma(v) & \text{if } v \in L(T), \\ \rho_S 1 & \text{if } v = \rho_T, \text{ and} \\ \rho_S i & \text{if } v = u_i \notin L(T), 1 \leq i \leq k. \end{cases}$$

With the help of Fig. 66, it is now easy to verify that (i) τ_T is a time map for T, (ii) the reconciliation map μ is time-consistent, and (iii) $\lambda_{\mathcal{S}} = \lambda$. In summary, $\mathcal{S} =$

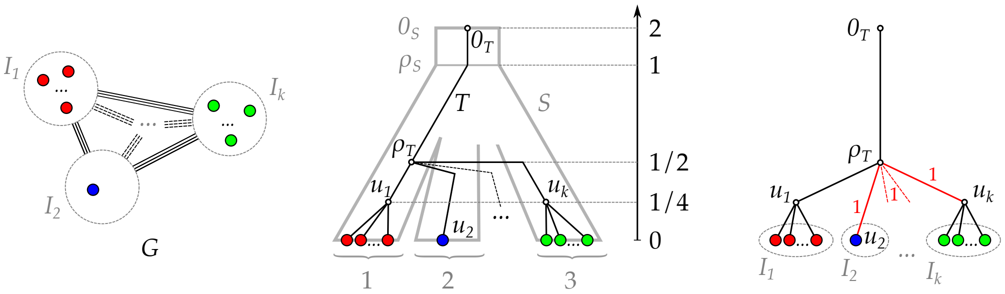


Fig. 66. Construction in the proof of Prop. 8.2.

$(T, S, \sigma, \mu, \tau_T, \tau_S)$ is a relaxed scenario, and $(G, \sigma) = (\mathscr{F}(\mathcal{S}), \sigma)$ is an rs-Fitch graph. □

Although every complete multipartite graph can be colored in such a way that it becomes an rs-Fitch graph (cf. Prop. 8.2), there are colored, complete multipartite graphs (G, σ) that are not rs-Fitch graphs, i.e., that do not derive from a relaxed scenario (cf. Thm. 8.3). We summarize this discussion in the following

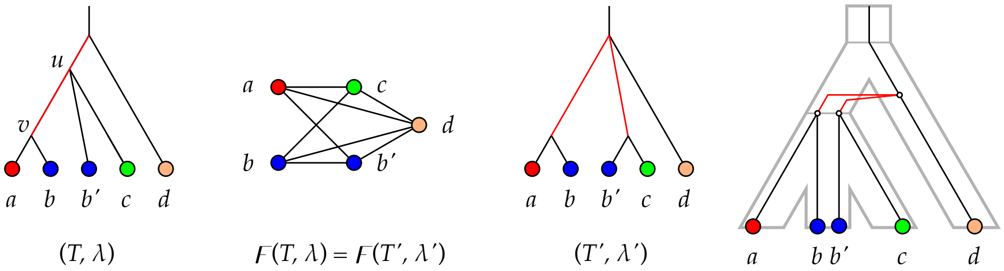
Observation 8.2. *There are (planted) 0/1-edge labeled trees* (T, λ) *and colorings* $\sigma\colon L(T) \to M$ *such that there is no relaxed scenario* $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ *with* $\lambda = \lambda_{\mathcal{S}}$.

A subtle – but important – observation is that trees (T, λ) with coloring σ for which Obs. 8.2 applies may still encode an rs-Fitch graph $(\mathscr{F}(T, \lambda), \sigma)$, see Example 8.1 and Fig. 67. The latter is due to the fact that $\mathscr{F}(T, \lambda) = \mathscr{F}(T', \lambda')$ may be possible for a different tree (T', λ') for which there is a relaxed scenario $\mathcal{S}' = (T', S, \sigma, \mu, \tau_T, \tau_S)$ with $\lambda' = \lambda_{\mathcal{S}}$. In this case, $(\mathscr{F}(T, \lambda), \sigma) = (\mathscr{F}(\mathcal{S}'), \sigma)$ is an rs-Fitch graph.

Example 8.1. *Consider the planted edge-labeled tree* (T, λ) *shown in Fig. 67 with leaf set* $L = \{a, b, b', c, d\}$*, together with a coloring* σ *where* $\sigma(b) = \sigma(b')$ *and* $\sigma(a), \sigma(b), \sigma(c), \sigma(d)$ *are pairwise distinct.*
Assume, for contradiction, that there is a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ *with* $(T, \lambda) = (T, \lambda_{\mathcal{S}})$*. Hence,* $\mu(v)$ *and* $\mu(b) = \sigma(b)$ *as well as* $\mu(u)$ *and* $\mu(b') = \sigma(b)$ *must be comparable in* S*. Therefore,* $\mu(u)$ *and* $\mu(v)$ *must both be comparable to* $\sigma(b)$ *and thus, they are located on the path from* ρ_S *to* $\sigma(b)$*. But this implies that* $\mu(u)$ *and* $\mu(v)$ *are comparable in* S*; a contradiction, since then* $\lambda_{\mathcal{S}}(u, v) = 0 \neq \lambda(u, v) = 1$.

8.2 LDT GRAPHS AND RS-FITCH GRAPHS

We proceed to investigate to what extent an LDT graph provides information about an rs-Fitch graph. As we shall see in Thm. 8.2, there is indeed a close connection between rs-Fitch graphs and LDT graphs. We start with a useful relation between the edges of rs-Fitch graphs and the reconciliation maps μ of their relaxed scenarios.



(T,λ) $\mathscr{F}(T,\lambda)=\mathscr{F}(T',\lambda')$ (T',λ')

Fig. 67. 0/1-edge-labeled tree (T,λ) for which no relaxed scenario exists such that $(T,\lambda)=(T,\lambda_{\mathcal{S}})$ (see Example 8.1). Red edges indicates 1-labeled edges. Nevertheless, for $\mathscr{F}:=\mathscr{F}(T,\lambda)$, there is an alternative tree (T',λ') for which a relaxed scenario $\mathcal{S}=(T',S,\sigma,\mu,\tau_T,\tau_S)$ exists (right) such that $\mathscr{F}=\mathscr{F}(T',\lambda')=\mathscr{F}(\mathcal{S})$.

Lemma 8.2. *Let $\mathscr{F}(\mathcal{S})$ be an rs-Fitch graph for some relaxed scenario $\mathcal{S}$. Then, $ab\notin E(\mathscr{F}(\mathcal{S}))$ implies that $\operatorname{lca}_S(\sigma(a),\sigma(b))\preceq_S\mu(\operatorname{lca}_T(a,b))$.*

Proof. Assume first that $ab\notin E(\mathscr{F}(\mathcal{S}))$ and denote by P_{xy} the unique path in T that connects the two vertices x and y. Clearly, $u:=\operatorname{lca}_T(a,b)$ is contained in P_{ab}, and this path P_{ab} can be subdivided into the two paths $P_{u,a}$ and $P_{u,b}$ that have only vertex u in common. Since $ab\notin E(\mathscr{F}(\mathcal{S}))$, none of the edges vw along the path P_{ab} in T is a transfer edge, and thus, the images $\mu(v)$ and $\mu(w)$ are comparable in S. This implies that the images of any two vertices along the path $P_{u,a}$ as well as the images of any two vertices along $P_{u,b}$ are comparable. In particular, therefore, $\mu(u)$ is comparable with both $\mu(a)=\sigma(a)=:A$ and $\mu(b)=\sigma(b)=:B$, where we may have $A=B$. Together with the fact that A and B are leaves in S, this implies that $\mu(u)$ is an ancestor of A and B. Since $\operatorname{lca}_S(A,B)$ is the "last" vertex that is an ancestor of both A and B, we have $\operatorname{lca}_S(A,B)\preceq_S\mu(u)$. ☐

The next result shows that a subset of transfer edges can be inferred immediately from LDT graphs:

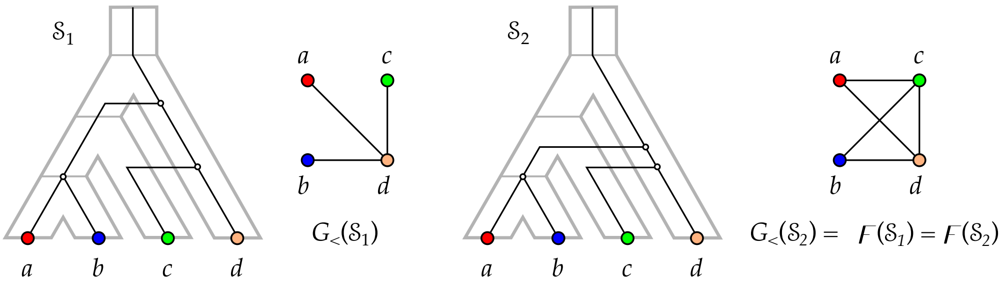
Theorem 8.1. *If (G,σ) is an LDT graph, then $G\subseteq\mathscr{F}(\mathcal{S})$ for all relaxed scenarios $\mathcal{S}$ that explain (G,σ).*

Proof. Let $\mathcal{S}=(T,S,\sigma,\mu,\tau_T,\tau_S)$ be a relaxed scenario that explains (G,σ), i.e., $G=G_<(\mathcal{S})$. By definition, $V(G)=V(\mathscr{F}(\mathcal{S}))=L(T)$. Hence it remains to show that $E(G)\subseteq E(\mathscr{F}(\mathcal{S}))$. To this end, consider $ab\in E(G)$ and assume, for contradiction, that $ab\notin E(\mathscr{F}(\mathcal{S}))$. Let $A:=\sigma(a)$ and $B:=\sigma(b)$. By Lemma 8.2, $\operatorname{lca}_S(A,B)\preceq_S\mu(\operatorname{lca}_T(a,b))$. But then, by Def. 7.1 and 7.2, $\tau_S(\operatorname{lca}_S(A,B))\leq\tau_S(\operatorname{lca}_T(a,b))$, implying $ab\notin E(G)$, a contradiction. ☐

Since we only have that xy is an edge in $\mathscr{F}(\mathcal{S})$ if the path connecting x and y in the tree T of $\mathcal{S}$ contains a transfer edge, Thm. 8.1 immediately implies

Corollary 8.1. *For every relaxed scenario $\mathcal{S}=(T,S,\sigma,\mu,\tau_T,\tau_S)$ without transfer edges, it holds that $E(G_<(\mathcal{S}))=\emptyset$.*

Thm. 8.1 provides the formal justification for indirect phylogenetic approaches to HGT inference that are based on the work of Lawrence and Hartl [186], Clarke et al. [55], and Novichkov et al. [229] by showing that $xy\in E(G_<(\mathcal{S}))$ can be explained only by HGT, irrespective of how complex the true biological scenario might have been. However, it does not cover



$\mathcal{S}_1$
a
b
c
d
a
c
b
d
$G_<(\mathcal{S}_1)$
$\mathcal{S}_2$
a
b
c
d
a
c
b
d
$G_<(\mathcal{S}_2) = \digamma(\mathcal{S}_1) = \digamma(\mathcal{S}_2)$

Fig. 68. Two relaxed scenarios $\mathcal{S}_1$ and $\mathcal{S}_2$ with the same rs-Fitch graph $\digamma = \digamma(\mathcal{S}_1) = \digamma(\mathcal{S}_2)$ (right) and different LDT graphs $G_<(\mathcal{S}_1) \neq \digamma$ and $G_<(\mathcal{S}_2) = \digamma$.

all HGT events. Fig. 68 shows that there are relaxed scenarios $\mathcal{S}$ for which $G_<(\mathcal{S}) \neq \digamma(\mathcal{S})$ even though $\digamma(\mathcal{S})$ is properly colored. Moreover, it is possible that an rs-Fitch graph (G, σ) contains edges $xy \in E(G)$ with $\sigma(x) = \sigma(y)$. In particular, therefore, an rs-Fitch graph is not always an LDT graph.

It is natural, therefore, to ask whether for every properly colored Fitch graph there is a relaxed scenario $\mathcal{S}$ such that $G_<(\mathcal{S}) = \digamma(\mathcal{S})$. An affirmative answer is provided by

Theorem 8.2. *The following statements are equivalent.*

1. *(G, σ) is a properly colored complete multipartite graph.*
2. *There is a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ with coloring σ such that $G = G_<(\mathcal{S}) = \digamma(\mathcal{S})$.*
3. *(G, σ) is complete multipartite and an LDT graph.*
4. *(G, σ) is properly colored and an rs-Fitch graph.*

In particular, for every properly colored complete multipartite graph (G, σ) the triple set $\mathfrak{S}(G, \sigma)$ is consistent.

Proof. (1) implies (2). We assume that (G, σ) is a properly colored multipartite graph and set $L := V(G)$ and $E := E(G)$. If $E = \emptyset$, then the relaxed scenario $\mathcal{S}$ constructed in the proof of Lemma 7.4 satisfies $G = G_<(\mathcal{S}) = \digamma(\mathcal{S})$, i.e., the graphs are edgeless. Hence, we assume that $E \neq \emptyset$ and explicitly construct a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ such that $G = G_<(\mathcal{S}) = \digamma(\mathcal{S})$.

The graph (G, σ) is properly colored and complete multipartite by assumption. Let $I_1, \dots, I_k$ denote the independent sets of G. Since $E \neq \emptyset$, we have $k > 1$. Since all $x \in I_i$ are adjacent to all $y \in I_j$, $i \neq j$ and (G, σ) is properly colored, it must hold that $\sigma(I_i) \cap \sigma(I_j) = \emptyset$. For a fixed i let $v_i^1, \dots v_i^{|I_i|}$ denote the elements in I_i.

We first start with the construction of the species tree S. First we add a planted root 0_S with child ρ_S. Vertex ρ_S has children $w_1, \dots, w_k$ where each w_j corresponds to one I_j. Note, $\sigma\colon L \to M$ may not be surjective, in which case we would add one additional child x to ρ_S for each color $x \in M \setminus \sigma(L)$.

If $|\sigma(I_j)| = 1$, then we identify the single color $x \in \sigma(I_j)$ with w_j. Otherwise, i.e., if $|\sigma(I_j)| > 1$, vertex w_j has as children the set $\mathrm{child}_S(w_j) = \sigma(I_j)$ which are leaves in S. See Fig. 69 for an illustrative example. Now we can choose the time map τ_S for S such $\tau_S(0_S) = 3$, $\tau_S(\rho_S) = 2$, $\tau_S(x) = 0$ for all $x \in L(S)$ and $\tau_S(x) = 1$ for all $x \in V^0(S) \setminus \{\rho_S\}$.

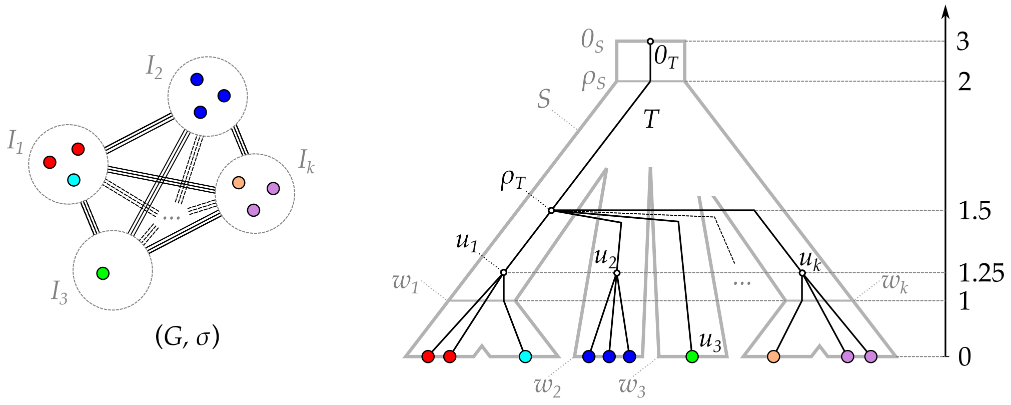


Fig. 69. Construction of the relaxed scenario $\mathcal{S}$ in the proof of Thm. 8.2.

We now construct T as follows. The tree T has planted root 0_T with child ρ_T. Vertex ρ_T has k children $u_1, \dots, u_k$ where each u_j corresponds to one I_j. Vertex u_j is a leaf if $|I_j| = 1$, and, otherwise, has exactly $|I_j|$ children that are uniquely identified with the elements in I_j.

We now define the time map τ_T and reconciliation map μ for $v \in V(T)$:

$$\tau_T(v) := \begin{cases} 3 = \tau_S(0_S) \\ 0 \\ 1.5 \\ 1.25 \end{cases} \qquad \mu(v) := \begin{cases} 0_S & \text{if } v = 0_T, \\ \sigma(v) & \text{if } v \in L(T), \\ \rho_S w_1 & \text{if } v = \rho_T, \text{ and} \\ \rho_S w_i & \text{if } v = u_i \notin L(T), 1 \leq i \leq k. \end{cases}$$

With the help of Fig. 69, it is now easy to verify that (i) τ_T is a time map for T, and that (ii) the reconciliation map μ is time-consistent. In summary, the constructed $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ is a relaxed scenario.

We continue with showing that $E = E(G_<(\mathcal{S})) = E(\digamma(\mathcal{S}))$. To this end, let $a, b \in L$ be two vertices. Note, $ab \in E$ if and only if $a \in I_i$ and $b \in I_j$ for distinct $i, j \in [k] := \{1, 2, \dots, k\}$.

First assume that $ab \in E$ and thus, $a \in I_i$ and $b \in I_j$ for distinct $i, j \in [k]$. By construction, $a \preceq_T u_i \neq u_j \succeq_T b$ with $\operatorname{lca}_T(u_i, u_j) = \rho_T$. In particular, we have $\mathsf{par}_T(u_i) = \mathsf{par}_T(u_j) = \rho_T$ and the path from a to b contains the two edges $\rho_T u_i$ and $\rho_T u_j$. By construction, we have $\mu(\rho_T) = \rho_S w_1)$ and for all $1 \leq l \leq k$, $\mu(u_l) = \sigma(u_l) = w_l$ if u_l is a leaf, and $\mu(u_l) = \rho_S w_l$ otherwise. These two arguments imply that $\mu(\rho_T)$ and $\mu(u_l)$ are comparable if and only if $u_l = u_1$. Now, since $u_i \neq u_j$, they cannot both be equal to u_1 and thus, at least one of the edges $\rho_T u_i$ and $\rho_T u_j$ is a transfer edge. Hence, $ab \in E(\digamma(\mathcal{S}))$. By construction, $ab \in E$ implies $\operatorname{lca}_T(a, b) = \rho_T$. Hence, we have $\mu(\operatorname{lca}_T(a, b)) = \mu(\rho_T) = \rho_S w_1 \prec_S \rho_S = \operatorname{lca}_S(\sigma(a), \sigma(b))$, and thus $ab \in E(G_<(\mathcal{S}))$.

Now assume that $ab \notin E$, and thus, $a, b \in I_i$ for some $i \in [k]$. It clearly suffices to consider the case $a \neq b$, and thus, $a, b \in \mathsf{child}_T(u_i)$ and $u_i \notin L(T)$ holds by construction. In particular, the path between a and b only consists of the edges $u_i a$ and $u_i b$. Moreover, we have $\sigma(a), \sigma(b) \preceq_S w_i$ and $\mu(u_i) = \rho_S w_i$. Hence, none of the edges $u_i a$ and $u_i b$ is a transfer edge, and $ab \notin E(\digamma(\mathcal{S}))$. We have $\mu(\operatorname{lca}_T(a, b)) = \rho_S w_i \succ_T w_i \succeq_T \operatorname{lca}_S(\sigma(a), \sigma(b))$, and thus $\tau_T(\operatorname{lca}_T(a, b)) > \tau_S(\operatorname{lca}_S(\sigma(a), \sigma(b)))$. Hence, $ab \notin E(G_<(\mathcal{S}))$.

In summary, $ab \in E$ if and only if $ab \in E(\digamma(\mathcal{S}))$ if and only if $ab \in E(G_<(\mathcal{S}))$, and consequently, $G = G_<(\mathcal{S}) = \digamma(\mathcal{S})$.

(2) implies (1). Thus, suppose that there is a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ such that $G = G_<(\mathcal{S}) = \digamma(\mathcal{S})$. Prop. 7.1 implies that $(G, \sigma) = (G_<(\mathcal{S}), \sigma)$ is properly colored. Moreover, $(G, \sigma) = (\digamma(\mathcal{S}), \sigma)$ is an rs-Fitch graph and thus, by Obs. 8.1, G is complete multipartite.

Statements (1) and (2) together with Prop. 8.1 imply (3). Conversely, if (3) is satisfied then Prop. 7.1 implies that (G, σ) is properly colored. This and the fact that G is complete multipartite implies (1). Therefore, Statements (1), (2) and (3) are equivalent.

Furthermore, (4) implies (1) by Obs. 8.1. Conversely, (G, σ) in Statement (2) is an rs-Fitch graph and an LDT graph. Hence it is properly colored by Prop. 7.1. Thus (2) implies (4).

Statement (3), in particular, implies that every properly colored complete multipartite (G, σ) is an LDT graph and, thus, there is a relaxed scenario $\mathcal{S}$ such that $G = G_<(\mathcal{S})$. Now, we can apply Lemma 7.6 to conclude that $\mathfrak{S}(G, \sigma)$ is consistent, which completes the proof. ☐

Corollary 8.2. *A colored graph (G, σ) is an LDT graph and an rs-Fitch graph if and only if (G, σ) is a properly colored complete multipartite graph (and thus, a properly colored Fitch graph for some 0/1-edge-labeled tree).*

Proof. If (G, σ) is an rs-Fitch graph then, by Obs. 8.1, G is a complete multipartite graph. Moreover, since (G, σ) is an LDT graph, (G, σ) is properly colored (cf. Prop. 7.1). Conversely, if (G, σ) is a properly colored complete multipartite graph it is, by Thm. 8.2(2), an rs-Fitch graph and an LDT graph. Now the equivalence between Statements (1) and (3) in Thm. 8.2 shows that (G, σ) is an LDT graph. ☐

Corollary 8.3. *Let (G, σ) be a vertex-colored graph. If $\mathfrak{T}(G) = \emptyset$ and $\mathfrak{S}(G, \sigma)$ is not consistent, then G is a complete multipartite graph (and thus, a Fitch graph for some 0/1-edge-labeled tree), but σ is not a proper vertex coloring of G.*

Proof. By definition, if $\mathfrak{T}(G) = \emptyset$, then G cannot contain an induced $K_2 + K_1$. By Prop. 8.1, G is a Fitch graph. Contraposition of the last statement in Thm. 8.2 and G being a Fitch graph for some (T, λ) implies that σ is not a proper vertex coloring of G. ☐

Relaxed scenarios for which $(\digamma(\mathcal{S}), \sigma)$ is properly colored do not admit two members of the same gene family that are separated by an HGT event. While restrictive, such models are not altogether unrealistic. Proper coloring of $(\digamma(\mathcal{S}), \sigma)$ is, in particular, the case if every horizontal transfer is *replacing*, i.e., if the original copy is effectively overwritten by homologous recombination [315], see also [52] for a detailed case study in *Streptococcus*. As a consequence of Thm. 8.2, LDT graphs are sufficient to describe replacing HGT. However, the incidence rate of replacing HGT decreases exponentially with phylogenetic distance between source and target [331], and additive HGT becomes the dominant mechanism between phylogenetically distant organisms. Still, replacing HGTs may also be the result of additive HGT followed by a loss of the (functionally redundant) vertically inherited gene.

8.3 RS-FITCH GRAPHS WITH GENERAL COLORINGS

In scenarios with additive HGT, the rs-Fitch graph is no longer properly colored and no-longer coincides with the LDT graph. Since not every vertex-colored complete multipartite graph (G,σ) is an rs-Fitch graph (cf. Thm. 8.3), we ask whether an LDT graph (G,σ) that is not itself already an rs-Fitch graph imposes constraints on the rs-Fitch graphs $(\digamma(\mathcal{S}),\sigma)$ that derive from relaxed scenarios $\mathcal{S}$ that explain (G,σ). As a first step towards this goal, we aim to characterize rs-Fitch graphs, i.e., to understand the conditions imposed by the existence of an underlying relaxed scenario $\mathcal{S}$ on the compatibility of the collection of independent sets $\mathcal{I}$ of G and the coloring σ. As we shall see, these conditions can be explained in terms of an auxiliary graph that we introduce in a very general setting:

Definition 8.4. *Let L be a set, $\sigma\colon L \to M$ a map and $\mathcal{I} = \{I_1,\dots,I_k\}$ a set of subsets of L. Then the graph $\mathcal{A}_\digamma(\sigma,\mathcal{I})$ has vertex set M and edges xy if and only if $x \neq y$ and $x,y \in \sigma(I')$ for some $I' \in \mathcal{I}$. We define an edge labeling $\ell : E(\mathcal{A}_\digamma(\sigma,\mathcal{I})) \to 2^{\mathcal{I}}$ such that $\ell(e) := \{I \in \mathcal{I} \mid \exists x,y \in I \text{ s.t. } \sigma(x)\sigma(y) = e\}$.*

By construction $\mathcal{A}_\digamma(\sigma,\mathcal{I}')$ is a subgraph of $\mathcal{A}_\digamma(\sigma,\mathcal{I})$ whenever $\mathcal{I}' \subseteq \mathcal{I}$. The labeling of an edge e records the sets $I \in \mathcal{I}$ that imply the presence of the edge. As it turns out, rs-Fitch graphs are characterized by the structure of their auxiliary graphs $\mathcal{A}_\digamma$ as shown in the next

Theorem 8.3. *A graph (G,σ) is an rs-Fitch graph if and only if (i) it is complete multipartite with independent sets $\mathcal{I} = \{I_1,\dots,I_k\}$, and (ii) if $k > 1$, there is an independent set $I' \in \mathcal{I}$ such that $\mathcal{A}_\digamma(\sigma,\mathcal{I} \setminus \{I'\})$ is disconnected.*

Proof. Let $G = (L,E)$ be a graph with coloring $\sigma\colon L \to M$. Suppose first that G satisfies (i) and (ii). To show that (G,σ) is an rs-Fitch graph, we will construct a relaxed scenario $\mathcal{S} = (T,S,\sigma,\mu,\tau_T,\tau_S)$ such that $G = \digamma(\mathcal{S})$. If $k = 1$, or equivalently $E = \emptyset$, then the relaxed scenario $\mathcal{S}$ constructed in the proof of Lemma 7.4 satisfies $G = \digamma(\mathcal{S})$, i.e., both graphs are edgeless. Now assume that $k > 1$ and thus, $E \neq \emptyset$. Hence, we can choose an independent set $I' \in \mathcal{I}$ such that $\mathcal{A}'_\digamma := \mathcal{A}_\digamma(\sigma,\mathcal{I} \setminus \{I'\})$ is disconnected. Note that $\mathcal{I} \setminus \{I'\}$ is non-empty since $k > 1$. Moreover, since $\mathcal{A}'_\digamma$ is a disconnected graph on the color set M, there is a connected component C of $\mathcal{A}'_\digamma$ such that $(M \setminus C) \cap \sigma(I') \neq \emptyset$. Hence $M_1 := M \setminus C$ and $M_2 := C$ form a bipartition of M such that neither M_1 nor M_2 are empty sets.

We continue by showing that every $I \in \mathcal{I} \setminus \{I'\}$ satisfies either $\sigma(I) \subseteq M_1$ or $\sigma(I) \subseteq M_2$. To see this, assume, for contradiction, that there are colors $A \in \sigma(I) \cap M_1$ and $B \in \sigma(I) \cap M_2$ for some $I \in \mathcal{I} \setminus \{I'\}$. Thus, $B \in C$ and, by definition, $AB \in E(\mathcal{A}'_\digamma)$. Therefore, A and B must lie in the connected component C; a contradiction. Therefore, we can partition $\mathcal{I} \setminus \{I'\}$ into $\mathcal{I}_1 := \{I \in \mathcal{I} \setminus \{I'\} \mid \sigma(I) \subseteq M_1\}$ and $\mathcal{I}_2 := \{I \in \mathcal{I} \setminus \{I'\} \mid \sigma(I) \subseteq M_2\}$. Note that one of the sets $\mathcal{I}_1$ and $\mathcal{I}_2$, but not both of them, may be empty. This may be the case, for instance, if σ is not surjective.

Now, we construct a relaxed scenario $\mathcal{S} = (T,S,\sigma,\mu,\tau_T,\tau_S)$ such that $G = \digamma(\mathcal{S})$. We first define the species tree S as the planted tree where ρ_S (i.e. the single child of 0_S) hast two children w_1 and w_2. If $|M_1| = 1$, we identify w_1 with the single element in M_1, and otherwise, we set $\mathsf{child}_S(w_1) = L(S(w_1)) := M_1$. We proceed analogously for w_2 and M_2. Thus, S is phylogenetic by construction. We choose the time map τ_S by putting $\tau_S(0_S) = 2$, $\tau_S(\rho_S) = 1$, $\tau_S(w_1) = \tau_S(w_2) = 0.5$ and $\tau_S(x) = 0$ for all $x \in L(S)$. This completes the construction of S and τ_S.

We proceed with the construction of the gene tree T, its time map τ_T and the reconciliation map μ. This tree T has leaf set L, planted root 0_T, and root ρ_T. We set $\mu(0_T) = 0_S$ and $\tau_T(0_T) = \tau_S(0_S) = 2$, and moreover $\mu(x) = \sigma(x)$ and $\tau_T(x) = 0$ for all $x \in L$.

For each $I_j \in \mathcal{I} \setminus \{I'\}$, we add a vertex u_j. We will later specify how these vertices are connected (via paths) to ρ_T. If $|I_j| = 1$, u_j becomes a leaf of T that is identified with the unique element in I_j. Otherwise, we add exactly $|I_j|$ children to u_j, each of which is identified with one of the elements in I_j. If u_j is a leaf, we already defined $\mu(u_j) = \sigma(u_j)$ and $\tau_T(u_j) = 0$.

Otherwise, we set $\tau_T(u_j) = 0.6$ and $\mu(u_j) = \rho_S w_1$ if $I_j \in \mathcal{I}_1$ and $\mu(u_j) = \rho_S w_2$ if $I_j \in \mathcal{I}_2$. Recall that $M_1 \cap \sigma(I') \neq \emptyset$. However, both $M_2 \cap \sigma(I') \neq \emptyset$ and $M_2 \cap \sigma(I') = \emptyset$ are possible. The latter case appears e.g. whenever $\mathcal{A}_{\mathcal{F}}(\sigma, \mathcal{I})$ was already disconnected. To connect the vertices u_j to ρ_T, we distinguish the three mutually exclusive cases:

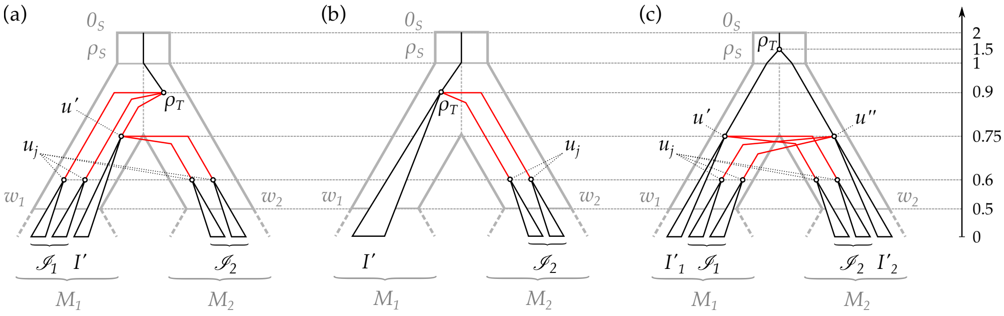


Fig. 70. Illustration of the relaxed scenario constructed in the *if*-direction of the proof of Thm. 8.3. For Cases (a) and (c), only the situation in which a vertex u' and u'', resp., is necessary is shown. Otherwise, the single element in I', I'_1 or I'_2 would be a child of the root ρ_T. Moreover, the vertices u_j are drawn under the assumption that $|I_j| > 1$. Otherwise, there are identified with the single leaf in I_j.

Case (a): $M_2 \cap \sigma(I') = \emptyset$ and $\mathcal{I}_1 \neq \emptyset$.
We set $\mu(\rho_T) = \rho_S w_2$ and $\tau_T(\rho_T) = 0.9$. We attach all u_j that correspond to elements $I_j \in \mathcal{I}_1$ as children of ρ_T. If $|I'| > 1$ or $\mathcal{I}_2 \neq \emptyset$, we create a vertex u' to which all elements in I' and all u_j such that $I_j \in \mathcal{I}_2$ are attached as children, attach u' as a child of ρ_T, and set $\mu(u') = \rho_S w_1$ and $\tau_T(u') = 0.75$. Otherwise, we simply attach the single element x' in I' as a child of ρ_T. Clearly, the so constructed tree T is phylogenetic. Note that the edges $\rho_T u_j$ with $I_j \in \mathcal{I}_1$ as well as the edges $u' u_j$ with $I_j \in \mathcal{I}_2$ are transfer edges. Together with $\rho_T u'$ or $\rho_T x$, respectively, these are the only transfer edges.
Case (b): $M_2 \cap \sigma(I') = \emptyset$ and $\mathcal{I}_1 = \emptyset$.
By the arguments above, the latter implies $\mathcal{I}_2 \neq \emptyset$. Hence, we can set $\mu(\rho_T) = \rho_S w_1$ and $\tau_T(\rho_T) = 0.9$ and attach all elements of I' as well as the vertices u_j corresponding to the independent sets $I_j \in \mathcal{I}_2 = \mathcal{I} \setminus \{I'\}$ as children of ρ_T. Since $|I'| \geq 1$ and $\mathcal{I}_2 \geq 1$, the tree T obtained in this manner is again phylogenetic. Moreover, note that the transfer edges are exactly the edges $\rho_T u_j$.
Case (c): $M_2 \cap \sigma(I') \neq \emptyset$.
In this case, the sets $I'_1 := \{x \in I' \mid \sigma(x) \in M_1\}$ and $I'_2 := \{x \in I' \mid \sigma(x) \in M_2\}$ must be non-empty. We set $\mu(\rho_T) = 0_T \rho_T$ and $\tau_T(\rho_T) = 1.5$. If $|I'_1| > 1$ or $\mathcal{I}_2 \neq \emptyset$, we create a vertex u' to which all elements in I'_1 and all u_j such that $I_j \in \mathcal{I}_2$ are attached as children, and set $\mu(u') = \rho_S w_1$ and $\tau_T(u') = 0.75$. Otherwise, we simply attach the single element in I'_1 as a child of ρ_T. For the "other side", we proceed analogously: If $|I'_2| > 1$ or $\mathcal{I}_1 \neq \emptyset$, we create a vertex u'' to which all elements in I'_2 and all u_j such that $I_j \in \mathcal{I}_1$ are attached as children, and set $\mu(u') = \rho_S w_2$ and $\tau_T(u'') = 0.75$. Otherwise, we simply attach the single element in I'_2 as a child of ρ_T. By construction, the so constructed tree is again phylogenetic. Moreover, the transfer edges are exactly the edges $u' u_j$ and $u'' u_j$.

Using Fig. 70, one can easily verify that, in all three Cases (a)-(c), the reconciliation map μ is time-consistent with τ_T and τ_S. Thus, $\mathcal{S}$ is a relaxed scenario. Moreover, Fig. 70 together with the fact that $\sigma(I) \subseteq M_1$ holds for all $I \in \mathcal{I}_1$, and $\sigma(I) \subseteq M_2$ holds for all $I \in \mathcal{I}_2$, shows that $G = \digamma(\mathcal{S})$ in all three cases. Hence, (G, σ) is an rs-Fitch graph.

For the *only-if*-direction, assume that $(G = (V, E), \sigma)$ is an rs-Fitch graph. Hence, there exists a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ such that $G = \digamma(\mathcal{S})$. By Obs. 8.1 and Prop. 8.1, (G, σ) is a complete multipartite graph that is determined by its set of independent sets $\mathcal{I} = \{I_1, \dots, I_k\}$. Hence, Condition (i) is satisfied.

Now assume, for contradiction, that Condition (ii) is violated. Thus $k \geq 2$ and there is no independent set $I' \in \mathcal{C}$ such that $\mathcal{A}_\digamma(\sigma, \mathcal{I} \setminus \{I'\})$ is disconnected. If $|M| = 1$, then the species tree S only consists of the planted root 0_S and the root ρ_S, which in this case is identified with the single element in M. Clearly, all vertices and edges are comparable in such a tree S, and hence, there is no transfer edges in $\mathcal{S}$, implying $E = \emptyset$ and thus $|\mathcal{I}| = 1$; a contradiction to $k \geq 2$.

Thus we have $|M| \geq 2$ and the root ρ_S of the species tree S has at least two children. Since $\mathcal{A}_\digamma(\sigma, \mathcal{I} \setminus \{I'\})$ is connected for every $I' \in \mathcal{C}$, the graph $\mathcal{A}_\digamma(\sigma, \mathcal{I})$ is also connected. Since each color appears at most once as a leaf of S, $\sigma(L(S(v_1))) \cap \sigma(L(S(v_2))) = \emptyset$ holds for any two distinct children $v_1, v_2 \in \mathsf{child}_S(\rho_S)$. These three assertions, together with the definition of the auxiliary graph $\mathcal{A}_\digamma(\sigma, \mathcal{I})$, imply that there are two distinct colors $A, B \in M$ such that AB is an edge in $\mathcal{A}_\digamma(\sigma, \mathcal{I})$, $A \preceq_S v_1$ and $B \prec_S v_2$ for distinct children $v_1, v_2 \in \mathsf{child}_S(\rho_S)$. By definition of $\mathcal{A}_\digamma(\sigma, \mathcal{I})$ there is an independent set $I' \in \mathcal{I}$ containing a vertex $a \in I'$ with $\sigma(a) = A$ and a vertex $b \in I'$ with $\sigma(b) = B$. Since a and b lie in the same independent set, we have $ab \notin E$. By Lemma 8.2, $\mu(\mathrm{lca}_T(a, b)) \succeq_S \mathrm{lca}_S(A, B) = \rho_S$. Since, by assumption, $\mathcal{A}_\digamma(\sigma, \mathcal{I} \setminus \{I'\})$ is also connected, we find two distinct colors C and D (not necessarily distinct from A and B) such that CD is an edge in $\mathcal{A}_\digamma(\sigma, \mathcal{I})$, $C \preceq_S v_3$ and $D \prec_S v_4$ for distinct children $v_3, v_4 \in \mathsf{child}_S(\rho_S)$ (but not necessarily distinct from v_1 and v_2), and in particular, an independent set $I'' \in \mathcal{I} \setminus \{I'\}$ containing a vertex $c \in I''$ with $\sigma(c) = C$ and a vertex $d \in I''$ with $\sigma(d) = D$. By construction, $I' \neq I''$, and thus, all edges between I' and I'' exist in G, in particular the edges ac, ad, bc, bd. Since $c, d \in I''$, we have $cd \notin E$ and thus, by Lemma 8.2, $\mu(\mathrm{lca}_T(c, d)) \succeq_S \mathrm{lca}_S(C, D) = \rho_S$.

We now consider the unique path P in T that connects $\mathrm{lca}_T(a, b)$ and $\mathrm{lca}_T(c, d)$. Since μ is time-consistent and $\mu(\mathrm{lca}_T(a, b)), \mu(\mathrm{lca}_T(c, d)) \succeq_S \rho_S$, we conclude that, for every edge uv along this path P, we have $\mu(u), \mu(v) \succeq_S \rho_S$ and thus $\mu(u), \mu(v) \in \{\rho_S, 0_S\rho_S\}$. But then, $\mu(u)$ and $\mu(v)$ are comparable in S. Therefore, P does not contain any transfer edge. Since $ab \notin E$, the path connecting a and $\mathrm{lca}_T(a, b)$ does not contain any transfer edges. Likewise, $cd \notin E$ implies that the path connecting c and $\mathrm{lca}_T(c, d)$ does not contain any transfer edges. Thus, the path connecting a and c also does not contain any transfer edge, which implies that $ac \notin E(\digamma(\mathcal{S})) = E$; a contradiction since a and c belong to two distinct independent sets.

Hence, we conclude that for $k > 1$ there exists an independent set $I' \in \mathcal{C}$ such that $\mathcal{A}_\digamma(\sigma, \mathcal{I} \setminus \{I'\})$ is disconnected. ☐

As a consequence of Thm. 8.3, we obtain

Corollary 8.4. *rs-Fitch graphs can be recognized in polynomial time.*

Proof. Every rs-Fitch graph (G, σ) must be complete multipartite, which can be verified in polynomial time. In this case, the set of independent sets $\mathcal{I} = \{I_1, \dots, I_k\}$ of G can also be determined and the graph $\mathcal{A}_\digamma(\sigma, \mathcal{I})$ can be constructed in polynomial time. Finally, we need to find an independent set $I' \in \mathcal{I}$, such that $\mathcal{A}_\digamma(\sigma, \mathcal{I} \setminus \{I'\})$ is disconnected. Clearly, checking whether $\mathcal{A}_\digamma(\sigma, \mathcal{I} \setminus \{I'\})$ is disconnected can be done in polynomial time and since there are at most $|V(G)|$ independent sets in $\mathcal{I}$, finding an independent set I' such that $\mathcal{A}_F(\sigma, \mathcal{I} \setminus \{I'\})$ is disconnected (if one exists) can be done in polynomial time as well. $\square$

Corollary 8.5. *Let (G, σ) be a complete multipartite graph with coloring $\sigma\colon V(G) \to M$ and set of independent sets $\mathcal{I}$. Then, (G, σ) is an rs-Fitch graph if and only if $\mathcal{A}_F(\sigma, \mathcal{I})$ is disconnected or there is a cut $Q \subseteq E(\mathcal{A}_F(\sigma, \mathcal{I}))$ such that all edges $e \in Q$ have the same label $\ell(e) = \{I\}$ for some $I \in \mathcal{I}$.*

Proof. If $\mathcal{A}_F(\sigma, \mathcal{I})$ is disconnected, then $\mathcal{A}_F(\sigma, \mathcal{I} \setminus \{I\})$ remains disconnected for all $I \in \mathcal{I}$ and, by Thm. 8.3, (G, σ) is an rs-Fitch graph.

If there is a cut $Q \subseteq E(\mathcal{A}_F(\sigma, \mathcal{I}))$ such that all edges $e \in Q$ have the same label $\ell(e) = \{I\}$ for some $I \in \mathcal{I}$, then, by definition, $E(\mathcal{A}_F(\sigma, \mathcal{I} \setminus \{I\})) \subseteq E' := E(\mathcal{A}_F(\sigma, \mathcal{I})) \setminus Q$. Since Q is a cut in $\mathcal{A}_F(\sigma, \mathcal{I})$, the resulting graph $\mathcal{A}'_F = (M, E')$ is disconnected. By the latter arguments, $\mathcal{A}_F(\sigma, \mathcal{I} \setminus \{I\})$ is a subgraph of $\mathcal{A}'_F$, and thus, disconnected as well. By Thm. 8.3, (G, σ) is an rs-Fitch graph.

Conversely, if (G, σ) is an rs-Fitch graph, then Thm. 8.3 implies that $\mathcal{A}_F(\sigma, \mathcal{I} \setminus \{I\})$ is disconnected for some $I \in \mathcal{I}$. If $\mathcal{A}_F(\sigma, \mathcal{I})$ was already disconnected, then there is nothing to show. Hence assume that $\mathcal{A}_F(\sigma, \mathcal{I}) = (M, E)$ is connected and let $\mathcal{A}_F(\sigma, \mathcal{I} \setminus \{I\}) = (M, E')$. Moreover, let $F \subseteq E$ be the subset of edges $e \in E$ with $I \in \ell(e)$. Note, F contains all edges of E that have potentially been removed from E to obtain E'. However, all edges $e = xy$ in F with $|\ell(e)| > 1$ must remain in $\mathcal{A}_F(\sigma, \mathcal{I} \setminus \{I\})$, since there is another independent set $I' \in \ell(e) \setminus \{I\}$ such that $x, y \in \sigma(I')$. Hence, only those edges e in F for which $|\ell(e)| = 1$ are removed from E. Hence, there is a cut $Q \subseteq F \subseteq E$ such that all edges $e \in Q$ have the same label $\ell(e) = \{I\}$ for some $I \in \mathcal{I}$. $\square$

Corollary 8.6. *If (G, σ) with coloring $\sigma\colon V(G) \to M$ is an rs-Fitch graph then there are no two disjoint independent sets I and I' of G with $\sigma(I) = \sigma(I') = M$.*

Proof. Let $\mathcal{I}$ be the set of independent sets of G. If $|\mathcal{I}| = 1$, there is nothing to show and thus, we assume that $|\mathcal{I}| > 1$. Assume, for contradiction, that there are two distinct independent sets $I, I' \in \mathcal{I}$ such that $\sigma(I) = \sigma(I') = M$. For every $I'' \in \mathcal{I}$, the set $\mathcal{I} \setminus \{I''\}$ clearly contains at least one of the two sets I and I', both of which contain all colors in M. Therefore, $\mathcal{A}_F(\sigma, \mathcal{I} \setminus \{I''\})$ is the complete graph by construction and, thus, connected for every $I'' \in \mathcal{I}$. This together with Thm. 8.3 implies that (G, σ) is not an rs-Fitch graph; a contradiction. $\square$

Corollary 8.7. *Every complete multipartite graph (G, σ) with a vertex coloring $\sigma\colon V(G) \to M$ that is not surjective is an rs-Fitch graph.*

Proof. If $\sigma\colon V(G) \to M$ is not surjective, then $\mathcal{A}_F(\sigma, \mathcal{I})$ is disconnected, where $\mathcal{I}$ denotes the set of independent sets of G. Hence, if $k > 1$, then $\mathcal{A}_F(\sigma, \mathcal{I} \setminus \{I\})$ remains disconnected for all $I \in \mathcal{I}$. By Thm. 8.3, (G, σ) is an rs-Fitch graph. $\square$

Cor. 8.7 may seem surprising since it implies that the property of being an rs-Fitch graph can depend on species (colors M) for which we have no genes L in the data. The reason is that an additional lineage in the species tree provides a place to "park" interior vertices in the gene tree from which HGT-edges can emanate that could not always be accommodated within lineages that have survivors – where they may force additional HGT edges.

Corollary 8.8. *Every Fitch graph (G, σ) that contains an independent set I and a vertex $x \in I$ with $\sigma(x) \notin \sigma(I')$ for all other independent sets $I' \neq I$, is an rs-Fitch graph.*

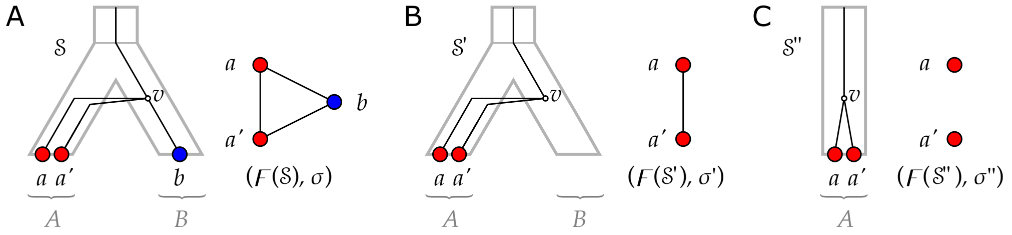


Fig. 71. Shown are three distinct relaxed scenarios $\mathcal{S}$, $\mathcal{S}'$ and $\mathcal{S}''$ with corresponding rs-Fitch graphs. Here $\sigma' = \sigma_{|\{a,a'\}}$ and $\sigma'' = \sigma_{|\{a,a'\},\{A\}}$ (cf. Def. 2.1). Putting $(G,\sigma) = (\digamma(\mathcal{S}),\sigma)$, one can observe that $(G[\{a,a'\}],\sigma') = (\digamma(\mathcal{S}'),\sigma')$ is an rs-Fitch graph. In contrast, σ'' is restricted to the "observable" part of species (consisting of A alone), and $(G[\{a,a'\}],\sigma'')$ is not an rs-Fitch graph, see text for further details.

Proof. Let $\mathcal{I}$ denote the set of independent sets of G. If there is an independent set $I \in \mathcal{I}$ that contains a vertex $x \in I$ with $\sigma(x) \notin \sigma(I')$ for all other independent sets $I' \neq I$, then the vertex $\sigma(x)$ in $\mathcal{A}_\digamma(\sigma, \mathcal{I} \setminus \{I\})$ is an isolated vertex and thus, $\mathcal{A}_\digamma(\sigma, \mathcal{I} \setminus \{I\})$ is disconnected. By Thm. 8.3, (G,σ) is an rs-Fitch graph. ☐

As for LDT graphs, the property of being an rs-Fitch graph is hereditary.

Corollary 8.9. *If $(G = (L,E),\sigma)$ is an rs-Fitch graph, then the colored vertex induced subgraph $(G[W],\sigma_{|W})$ is an rs-Fitch graph for all non-empty subsets $W \subseteq L$.*

Proof. It suffices to show the statement for $W = L \setminus \{x\}$ for an arbitrary vertex $x \in L$. If $G = (L,E)$ is edgeless, then $G[W]$ is edgeless and thus, by Thm. 8.3, an rs-Fitch graph.

Thus, assume that $E \neq \emptyset$ and thus, for the set $\mathcal{I}$ of independent sets of G it holds that $|\mathcal{I}| > 1$. Since G does not contain an induced $K_2 + K_1$, it is easy to see that $G[W]$ cannot contain an induced $K_2 + K_1$ and thus, $G[W]$ is a complete multipartite graph. Hence, Thm. 8.3(i) is satisfied. Moreover, if for the set $\mathcal{I}'$ of independent sets of $G[W]$ it holds that $|\mathcal{I}'| = 1$ then, Thm. 8.3 already shows that $(G[W],\sigma_{|W})$ is an rs-Fitch graph.

Thus, assume that $|\mathcal{I}'| > 1$. Now compare the labeling ℓ of the edges in $\mathcal{A}_\digamma = \mathcal{A}_\digamma(\sigma,\mathcal{I})$ and the labeling ℓ' of the edges in $\mathcal{A}'_\digamma = \mathcal{A}_\digamma(\sigma_{|W},\mathcal{I}')$. Note, $\mathcal{A}_\digamma$ and $\mathcal{A}'_\digamma$ have still the same vertex set M. Let $I \in \mathcal{I}$ with $x \in I$. For all vertices $y \in I$ with $\sigma(x) \neq \sigma(y)$, we have an edge $e = \sigma(x)\sigma(y)$ in $\mathcal{A}_\digamma$ and $I \in \ell(e)$. Consequently, for all edges e of $\mathcal{A}_\digamma$ that are present in $\mathcal{A}'_\digamma$ we have $\ell'(e) \subseteq \ell(e)$. In particular, $\mathcal{A}'_\digamma$ cannot have edges that are not present in $\mathcal{A}_\digamma$, since we reduced for one independent set the size by one. Therefore, $\mathcal{A}'_\digamma$ is a subgraph of $\mathcal{A}_\digamma$.

By Thm. 8.3, there is an independent set $I' \in \mathcal{I}$, not necessarily distinct from I, such that $\mathcal{A}_\digamma(\sigma,\mathcal{I} \setminus \{I'\})$ is disconnected. If $I' = \{x\}$, then $\mathcal{I}' = \mathcal{I} \setminus \{I'\}$ and $\mathcal{A}'_\digamma = \mathcal{A}_\digamma$ must be disconnected as well. Otherwise, $\mathcal{A}'_\digamma \subseteq \mathcal{A}_\digamma$ and similar arguments as above show that $\mathcal{A}_\digamma(\sigma,\mathcal{I}' \setminus \{I'\}) \subseteq \mathcal{A}_\digamma(\sigma,\mathcal{I} \setminus \{I'\})$. Therefore, in both of the latter cases, $\mathcal{A}_\digamma(\sigma,\mathcal{I}' \setminus \{I'\})$ is disconnected and Thm. 8.3 implies that $(G[W],\sigma_{|W})$ is an rs-Fitch graph. ☐

Note, however, that Cor. 8.9 is not satisfied if we restrict the codomain of σ to the observable part of colors, i.e., if we consider $\sigma_{|W,\sigma(W)} \colon W \to \sigma(W)$ instead of $\sigma_{|W} \colon W \to M$, even if σ is surjective. To see this consider the vertex colored graph (G,σ) with $V(G) = \{a,a',b\}$, $E(G) = \{aa',ab,a'b\}$ and

$\sigma\colon V(G) \to M = \{A, B\}$ where $\sigma(a) = \sigma(a') = A \neq \sigma(b) = B$. A possible relaxed scenario $\mathcal{S}$ for (G, σ) is shown in Fig. 71(A). The deletion of b yields $W = V(G) \setminus \{b\} = \{a, a'\}$ and the graph $(G[W], \sigma_{|W})$ for which $\mathcal{S}'$ with HGT-labeling $\lambda_{\mathcal{S}'}$ as in Fig. 71(B) is a relaxed scenario that satisfies $G[W] = \digamma(T, \lambda_{\mathcal{S}'})$. However, if we restrict the codomain of σ to obtain $\sigma_{|W,\{A\}}\colon \{a, a'\} \to \sigma(W) = \{A\}$, then there is no relaxed scenario $\mathcal{S}$ for which $G[W] = \digamma(T, \lambda_{\mathcal{S}})$, since there is only a single species tree S on $L(S) = \{A\}$ (Fig. 71(C)) that consists of the single edge $0_T A$ and thus, $\mu(v)$ and $\mu(a)$ as well as $\mu(v)$ and $\mu(a')$ must be comparable in this scenario.

8.4 LEAST RESOLVED TREES FOR FITCH GRAPHS

It is important to note that the characterization of rs-Fitch graphs in Thm. 8.3 does not provide us with a characterization of rs-Fitch graphs that share a common relaxed scenario with a given LDT graph. As a potential avenue to address this problem, we investigate the structure of least-resolved trees for Fitch graphs as possible source of additional constraints.

All trees considered in this section are rooted and phylogenetic but not planted unless stated differently. This is no loss of generality, since we are interested in Fitch-least-resolved trees, which are never planted because the edge incident with the planted root can be contracted without affecting the paths between the leaves.

Definition 8.5. *The edge-labeled tree (T, λ) is* Fitch-least-resolved *w.r.t. $\digamma(T, \lambda)$, if for all trees $T' \neq T$ that are displayed by T and every labeling λ' of T' it holds that $\digamma(T, \lambda) \neq \digamma(T', \lambda')$.*

Definition 8.6. *Let (T, λ) be an edge-labeled tree and let $e = xy \in E(T)$ be an inner edge. The tree $(T_{/e}, \lambda_{/e})$ with $L(T_{/e}) = L(T)$, is obtained by contraction of the edge e in T and by keeping the edge labels of all non-contracted edges.*

Note, if e is an inner edge of a phylogenetic tree T, then the tree $T_{/e}$ is again phylogenetic.

Definition 8.7. *An edge e in (T, λ) is* relevantly-labeled *in (T, λ) if, for the tree (T, λ') with $\lambda'(f) = \lambda(f)$ for all $f \in E(T) \setminus \{e\}$ and $\lambda'(e) \neq \lambda(e)$, it holds that $\digamma(T, \lambda) \neq \digamma(T, \lambda')$.*

Lemma 8.3. *An outer 0-edge $e = vx$ in (T, λ) is* relevantly-labeled *in (T, λ) if and only if $zx \notin E(\digamma(T, \lambda))$ for some $z \in L(T) \setminus \{x\}$.*

Proof. Assume that $e = vx$ is a relevantly-labeled outer 0-edge. Hence, for (T, λ') with $\lambda'(f) = \lambda(f)$ for all $f \in E(T) \setminus \{e\}$ and $\lambda'(e) = 1$, it holds that $\digamma(T, \lambda) \neq \digamma(T, \lambda')$. Since we only changed the label of the outer edge vx, it still holds that $yy' \in E(\digamma(T, \lambda'))$ if and only if $yy' \in E(\digamma(T, \lambda))$ for all distinct $y, y' \in L(T) \setminus \{x\}$. Moreover, since $\lambda'(e) = 1$ and $e = vx$ is an outer edge, we have $xz \in E(\digamma(T, \lambda'))$ for all $z \in L(T) \setminus \{x\}$. Thus, $\digamma(T, \lambda) \neq \digamma(T, \lambda')$ implies that $xz \notin E(\digamma(T, \lambda))$ for at least one $z \in L(T) \setminus \{x\}$.

Now, suppose that $zx \notin E(\digamma(T, \lambda))$ for some $z \in L(T) \setminus \{x\}$. Clearly, this implies that the outer edges $e = vx$ and $f = wz$ must be 0-edges and changing one of them to a 1-edge would imply that xz becomes an edge in the Fitch graph. Hence, e is relevantly-labeled in (T, λ). $\square$

Lemma 8.4. *For every tree (T, λ) and every inner 0-edge e of T, it holds $\mathcal{F}(T, \lambda) = \mathcal{F}(T_{/e}, \lambda_{/e})$.*

Proof. Suppose that (T, λ) contains an inner 0-edge $e = uv$. The contraction of this edge does not change the number of 1-edges along the paths connecting any two leaves. It affects the least common ancestor of x and y, if $\operatorname{lca}_T(x, y) = u$ or $\operatorname{lca}_T(x, y) = v$. In either case, however, the number of 1-edges between $\operatorname{lca}_T(x, y)$ and the leaves x and y remains unchanged. Hence, we have $\mathcal{F}(T, \lambda) = \mathcal{F}(T_{/e}, \lambda_{/e})$. ☐

Lemma 8.5. *If (T, λ) is a Fitch-least-resolved tree w.r.t. $\mathcal{F}(T, \lambda)$, then it does neither contain inner 0-edges nor inner 1-edges that are not relevantly-labeled.*

Proof. Suppose first, by contraposition, that (T, λ) contains an inner 0-edge $e = uv$. By Lemma 8.4, $\mathcal{F}(T, \lambda) = \mathcal{F}(T_{/e}, \lambda_{/e})$, and thus, (T, λ) is not Fitch-least-resolved.

Assume now, by contraposition, that (T, λ) contains an inner 1-edge e that is not relevantly-labeled. Hence, we can put $\lambda'(e) = 0$ and $\lambda(f) = \lambda(f')$ for all $f \in E(T) \setminus \{e\}$ and obtain $\mathcal{F}(T, \lambda) = \mathcal{F}(T, \lambda')$. Since (T, λ') contains an inner 0-edge, it cannot be Fitch-least-resolved. Therefore and by definition, (T, λ) cannot be Fitch-least-resolved as well. ☐

The converse of Lemma 8.5 is, however, not always satisfied. To see this, consider the Fitch graph $G \simeq K_3$ with vertices x, y and z. Now, consider the tree (T, λ) where T is the triple $xy|z$, the two outer edges incident to y and z are 0-edges while the remaining two edges in T are 1-edges. It is easy to verify that $G = \mathcal{F}(T, \lambda)$. In particular, the inner edge e is relevantly-labeled, since if $\lambda'(e) = 0$ we would have $yz \notin E(\mathcal{F}(T, \lambda'))$. However, (T, λ) is not Fitch-least-resolved w.r.t. G, since the star tree T' on the three leaves x, y, z is displayed by T, and the labeling λ' with $\lambda'(e) = 1$ for all $e \in E(T')$ provides a tree (T', λ') with $G = \mathcal{F}(T', \lambda')$.

Lemma 8.6. *A tree (T, λ) is a Fitch-least-resolved tree w.r.t. $\mathcal{F}(T, \lambda)$ if and only if $\mathcal{F}(T, \lambda) \neq \mathcal{F}(T_{/e}, \lambda')$ holds for all labelings λ' of $T_{/e}$ and all inner edges e in T.*

Proof. Let (T, λ) be an edge-labeled tree. Suppose first that (T, λ) is Fitch-least-resolved w.r.t. $\mathcal{F}(T, \lambda)$. For every inner edge e in T, the tree $T_{/e} \neq T$ is displayed by T. By definition of Fitch-least-resolved trees, we have $\mathcal{F}(T, \lambda) \neq \mathcal{F}(T_{/e}, \lambda')$ for every labeling λ' of $T_{/e}$.

For the converse, assume, for contraposition, that (T, λ) is not Fitch-least-resolved w.r.t. $\mathcal{F}(T, \lambda)$. Hence, there is a tree (T', λ') such that $T' \neq T$ is displayed by T and $\mathcal{F}(T, \lambda) = \mathcal{F}(T', \lambda')$. Clearly, T and T' must have the same leaf set. Therefore and since $T' < T$, the tree T' can be obtained from T by a sequence of contractions of inner edges $e_1, \dots, e_\ell$ (in this order) where $\ell \geq 1$. If $\ell = 1$, then we have $T' = T_{/e_1}$ and, by assumption, $\mathcal{F}(T, \lambda) = \mathcal{F}(T_{/e_1}, \lambda')$. Thus, we are done. Now assume $\ell \geq 2$. We consider the tree $(T_{/e_1}, \lambda'')$ where $\lambda''(f) = \lambda'(f)$ if $f \in E(T')$ and $\lambda''(f) = 0$ otherwise. Hence, (T', λ') can be obtained from $(T_{/e_1}, \lambda'')$ by stepwise contraction of the 0-edges $e_2, \dots, e_\ell$, and by keeping the labeling of λ'' for the remaining edges in each step. Hence, we can repeatedly apply Lemma 8.4 to conclude that $\mathcal{F}(T_{/e_1}, \lambda'') = \mathcal{F}(T', \lambda')$. Together with $\mathcal{F}(T, \lambda) = \mathcal{F}(T', \lambda')$, we obtain $\mathcal{F}(T, \lambda) = \mathcal{F}(T_{/e_1}, \lambda'')$, which completes the proof. ☐

As a consequence of Lemma 8.6, it suffices to show that $\mathcal{F}(T, \lambda) = \mathcal{F}(T_{/e}, \lambda')$ for some inner edge $e \in E(T)$ and some labeling λ' for $T_{/e}$ to show that (T, λ) is not Fitch-least-resolved tree w.r.t. $\mathcal{F}(T, \lambda)$. The next result characterizes Fitch-least-resolved trees and is very similar to the results for "directed" Fitch graphs of 0/1-edge-labeled trees (cf. Lemma 11(1,3) in [101]). However, we note that we defined Fitch-least-resolved in terms of all possible labelings λ' for trees T' displayed by T, whereas Geiß et al. [101] call (T,λ) least-resolved whenever $(T_{/e},\lambda_{/e})$ results in a (directed) Fitch graph that differs from the one provided by (T,λ) for every $e \in E(T)$.

Theorem 8.4. *Let G be a Fitch graph, and (T,λ) be a tree such that $G = \digamma(T,\lambda)$. If all independent sets of G are of size one (except possibly for one independent set), then (T,λ) is Fitch-least-resolved for G if and only if it is a star tree.*
If G has at least two independent sets of size at least two, then (T,λ) is Fitch-least-resolved for G if and only if

- *(a) every inner edge of (T,λ) is a 1-edge,*
- *(b) for every inner vertex $v \in V^0(T)$ there are (at least) two relevantly-labeled outer 0-edges vx, vy in (T,λ)*

In particular, if distinct $x,y \in L(T)$ are in the same independent set of G, then they have the same parent in T and $\mathsf{par}_T(x)x$, $\mathsf{par}_T(x)y$ are relevantly-labeled outer 0-edges.

Proof. Suppose that every independent set of G is of size one (except possibly for one). Let (T,λ) be the star tree where $\lambda(\rho_T v) = 1$ if and only if v is the single element in an independent set of size one. It is now a simple exercise to verify that $G = \digamma(T,\lambda)$. Since (T,λ) is a star tree, it is clearly Fitch-least-resolved. The converse follows immediately from this construction together with fact that the star tree is displayed by all trees with leaf set $V(G)$. In the following we assume that G contains at least two independent sets of size at least two.

First suppose that (T,λ) is Fitch-least resolved w.r.t. $\digamma(T,\lambda)$. By Lemma 8.5, Condition (a) is satisfied. We continue with showing that Condition (b) is satisfied. In particular, we show first that every inner vertex $v \in V^0(T)$ is incident to at least one relevantly-labeled outer 0-edge. To this end, assume, for contradiction, that (T,λ) contains an inner vertex $v \in V^0(T)$ for which this property is not satisfied.

That is, v is either (i) incident to 1-edges only (incl. $\lambda(\mathsf{par}_T(v)v) = 1$ in case $v \neq \rho_T$ by Condition (a)) or (ii) there is an outer 0-edge vx that is not relevantly-labeled. In Case (i), we put $\lambda' = \lambda$. In Case (ii), we obtain a new labeling λ' by changing the label of every outer 0-edge vx with $x \in \mathsf{child}_T(v) \cap L(T)$ to "1" while keeping the labels of all other edges. This does not affect the Fitch graph, since every such 0-edge is not relevantly-labeled, and thus, $zx \in E(\digamma(T,\lambda))$ for all $z \in L(T) \setminus \{x\}$ by Lemma 8.3. Hence, for both Cases (i) and (ii), for the labeling λ' *all* outer edges vx with $x \in \mathsf{child}_T(v) \cap L(T)$ are labeled as 1-edges, v is incident to 1-edges only (by Condition (a)) and $\digamma(T,\lambda) = \digamma(T,\lambda')$. We thus have $xy \in E(\digamma(T,\lambda')) = E(\digamma(T,\lambda))$ for all $x \in L(T(v))$ and $y \in L(T) \setminus L(T(v))$. Now, if $v \neq \rho_T$ let $e = uv := \mathsf{par}_T(v)v$. Otherwise, if $v = \rho_T$ then let $e = vu$ for some inner vertex $u \in \mathsf{child}_T(v)$. Note, such an inner edge $\rho_T u$ exists since G contains at least two independent sets of size at least two and T is not a star tree as shown above. Now consider the tree $(T_{/e},\lambda'_{/e})$, and denote by w the vertex obtained by contraction of the inner edge e. By construction, every path in $T_{/e}$ connecting any $x \in L(T(v))$ and $y \in L(T) \setminus L(T(v))$ must contain some 1-edge ww' with $w' \in \mathsf{child}_{T_{/e}}(w) = \mathsf{child}_T(v)$ implying $xy \in E(\digamma(T_{/e},\lambda'_{/e}))$. Moreover, the edge contraction does not affect whether or not the path between any vertices within $L(T(v))$ or within $L(T) \setminus L(T(v))$ contains a 1-edge. Hence, $\digamma(T,\lambda) = \digamma(T,\lambda') = \digamma(T_{/e},\lambda'_{/e})$, and (T,λ) is not Fitch-least-resolved; a contradiction. In summary, every inner vertex v must be incident to at least one relevantly-labeled outer 0-edge vx. By Lemma 8.3, vx is a relevantly-labeled outer 0-edge if and only if there is a vertex $z \in L(T) \setminus \{x\}$ such that $zx \notin E(\digamma(T,\lambda))$. By Condition (a), all inner edges in (T,λ) are 1-edges, and thus, there is only one place where the leaf z can be located in T, namely as a leaf adjacent to v. In particular, the outer edge vz is a relevantly-labeled 0-edge, since $zx \notin E(\digamma(T,\lambda))$. Therefore, Condition (b) is satisfied for every inner vertex v of T.

The latter arguments also show that all distinct vertices $x, y \in L(T)$ that are contained in the same independent set must have the same parent. Clearly, $\mathsf{par}_T(x)x$, $\mathsf{par}_T(x)y$ must be outer 0-edges, since otherwise $xy \in E(\digamma(T,\lambda))$. Hence, the final statement of the theorem is satisfied.

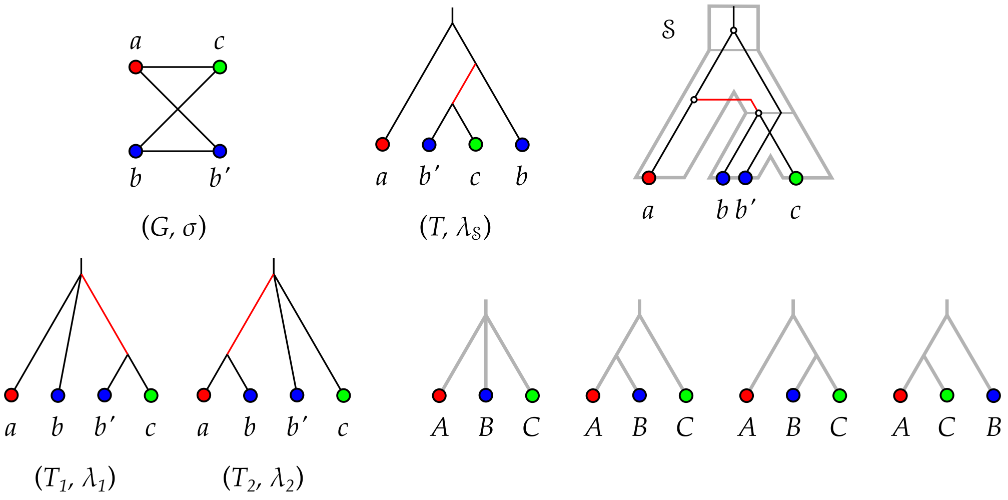
Now let (T,λ) be such that Conditions (a) and (b) are satisfied. First observe that none of the outer edges can be contracted without changing $L(T)$. Now let $e = uv$ be an inner edge. By Condition (a), e is a 1-edge. Moreover, by Condition (b), vertex u and v are both incident to at least two relevantly-labeled outer 0-edges. Hence, there are outer 0-edges ux, ux', vy, vy' with pairwise distinct leaves x, x', y, y' in T. Since uv is a 1-edge, we have $xy, xy', x'y, x'y' \in E(\digamma(T,\lambda))$. Moreover, we have $xx', yy' \notin E(\digamma(T,\lambda))$. Now consider the tree $(T_{/e}, \lambda')$ with an arbitrary labeling λ' and denote by w the vertex obtained by contraction of the inner edge uv. In this tree, x, x', y, y' all have the same parent w. If $\lambda'(wx) = 1$ *or* $\lambda'(wy) = 1$, we have $xx' \in \digamma(T_{/e}, \lambda')$ or $yy' \in E(\digamma(T_{/e}, \lambda'))$, respectively. If $\lambda'(wx) = 0$ *and* $\lambda'(wy) = 0$, we have $xy \notin E(\digamma(T_{/e}, \lambda'))$. Hence, it holds $\digamma(T_{/e}, \lambda') \neq \digamma(T,\lambda)$ in both cases. Since the inner edge e and λ' were chosen arbitrarily, we can apply Lemma 8.6 to conclude that (T,λ) is Fitch-least-resolved. ☐

As a consequence of Thm. 8.4, Fitch-least-resolved trees can be constructed in polynomial time. To be more precise, if a Fitch graph G contains only independent sets of size one (except possibly for one), we can construct a star tree T with edge labeling λ as specified in the proof of Thm. 8.4 to obtain the 0/1-edge labeled tree (T,λ) that is Fitch-least-resolved w.r.t. G. This construction can be done in $O(|V(G)|)$ time.

Now, assume that G has at least two independent sets of size at least two. Let $\mathcal{I}$ be the set of independent sets of G and $I_1, \dots, I_k \in \mathcal{I}$, $k \geq 2$ be all independent sets of size at least two. We now construct a tree (T,λ) with root ρ_T as follows: First we add k vertices $v_1 = \rho_T$ and $v_2, \dots, v_k$, and add inner edges $e_i = v_i v_{i+1}$ with label $\lambda(e_i) = 1$, $1 \leq i \leq k-1$. Each vertex v_i gets as children the leaves in I_i, $1 \leq i \leq k$ and all these additional outer edges obtain label "0". Finally, all elements in the remaining independent sets $\mathcal{I} \setminus \{I_1, \dots, I_k\}$ are of size one and are connected as leaves via outer 1-edges to the root $v_1 = \rho_T$. It is an easy exercise to verify that T is a phylogenetic tree and that $\digamma(T,\lambda) = G$. In particular, Thm. 8.4 implies that (T,λ) is Fitch-least-resolved w.r.t. G. This construction can be done in $O(|V(G)|)$ time. We summarize this discussion as

Proposition 8.3. *For a given Fitch graph G, a Fitch-least-resolved tree can be constructed in $O(|V(G)|)$ time.*

Fitch-least-resolved trees, however, are only of very limited use for the construction of relaxed scenarios $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ from an underlying Fitch graph. First note that we would need to consider *planted versions* of Fitch-least-resolved trees, i.e., Fitch-least-resolved trees to which a planted root is added, since otherwise, such trees cannot be part of an explaining scenario,



(G,σ) $(T,\lambda_{\mathcal{S}})$

(T_1,λ_1) (T_2,λ_2)

Fig. 72. An rs-Fitch graph (G,σ) and a possible relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ with $G = \digamma(T, \lambda_{\mathcal{S}})$. For the planted versions (T_1, λ_1) and (T_2, λ_2) of the Fitch-least-resolved trees of (G,σ) there is no relaxed scenario $\mathcal{S}$ such that $(T_i, \lambda_i) = (T_i, \lambda_{\mathcal{S}})$, $i \in \{1,2\}$. Red edges indicate 1-labeled (i.e., transfer) edges. See Example 8.2 for further details.

which is defined in terms of planted trees. Even though (G,σ) is an rs-Fitch graph, Example 8.2 shows that it is possible that there is no relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ with HGT-labeling $\lambda_{\mathcal{S}}$ such that $(T,\lambda) = (T, \lambda_{\mathcal{S}})$ for the planted version (T,λ) of *any* of its Fitch-least-resolved trees.

Example 8.2. *Consider the rs-Fitch graph (G,σ) with $V(G) = \{a, b, b', c\}$, $E(G) = \{ab', ac, bb', bc\}$ and surjective coloring σ such that $\sigma(a) = A$, $\sigma(b) = \sigma(b') = B$, $\sigma(c) = C$ and A, B, C are pairwise distinct. The rs-Fitch graph (G,σ), a Fitch tree (T,λ) and relaxed scenario $\mathcal{S}$ with $(T,\lambda) = (T,\lambda_{\mathcal{S}})$ as well as the planted versions (T_1,λ_1) and (T_2,λ_2) of its two Fitch-least-resolved trees are shown in Fig. 72.*

Fitch-least-resolved trees for (G,σ) must contain an inner 1-edge, since G has two independent sets of size two and by Thm. 8.4. Thus, it is easy to verify that there are no other Fitch-least-resolved trees for (G,σ).

By Lemma 8.2, we obtain $\operatorname{lca}_S(A,B) \preceq_S \mu(\operatorname{lca}_{T_i}(a,b))$ and $\operatorname{lca}_S(B,C) \preceq_S \mu(\operatorname{lca}_{T_i}(b',c))$, $i \in \{1,2\}$, for both (planted versions of the) Fitch-least-resolved trees. However, for all of the possible species trees on three leaves A, B, C, this implies that the images $\mu(\operatorname{lca}_{T_i}(a,b))$ and $\mu(\operatorname{lca}_{T_i}(b',c))$ are the single inner edge or the edge $0_T\rho_T$ in S. Therefore, $\mu(\operatorname{lca}_{T_i}(a,b))$ and $\mu(\operatorname{lca}_{T_i}(b',c))$ are always comparable in S. Hence, for all possible relaxed scenarios $\mathcal{S}$, we have $\lambda_{\mathcal{S}}(e) = 0$ for the single inner edge e, whereas $\lambda_i(e) = 1$ in T_i, $i \in \{1,2\}$. This implies that there is no relaxed scenario $\mathcal{S}$ with $(T_i,\lambda_i) = (T_i,\lambda_{\mathcal{S}})$, $i \in \{1,2\}$.

A slightly different mathematical question is left open here. Regarding 0/1-edge labeled (planted) trees (T,λ), it would be of interest to know whether there is always a relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ such that $(T,\lambda) = (T,\lambda_{\mathcal{S}'})$ for a suitable choice of σ. Elaborating on Thm. 8.2, it would be interesting to characterize the leaf colorings σ for (T, λ) such that there is a relaxed scenario $\mathcal{S}$ with $\digamma(T, \lambda) = \digamma(\mathcal{S})$.

8.5 EDITING PROBLEMS

8.5.1 *Editing Colored Graphs to rs-Fitch Graphs*

Even through at present it remains unclear whether rs-Fitch graphs can be estimated directly, the corresponding graph modification problems are at least of theoretical interest.

Problem 8.1 (rs-Fitch Graph Modification (rsF-M))**.**

Input:	*A colored graph* $(G = (V, E), \sigma)$ *and an integer* k*.*
Question:	*Is there a subset* $F \subseteq \binom{V}{2}$ *such that* $\lvert F\rvert \leq k$ *and* $(G \odot F, \sigma)$ *is an rs-Fitch graph where* $\odot \in \{\triangle, +, -\}$*?*

Since rs-Fitch graphs are complete multipartite graphs, their complements are disjoint unions of complete graphs. The problems rsF-M are thus closely related the cluster graph modification problems. Both Cluster Deletion and Cluster Editing are NP-complete, while Cluster Completion is polynomial (by completing each connected component to a clique, i.e., computing the transitive closure) [275]. We obtain

Theorem 8.5. rsF-C *and* rsF-E *are NP-complete.*

Proof. Since rs-Fitch graphs can be recognized in polynomial time, a given solution can be verified as being a yes- or no-answer in polynomial time. Thus, rsF-C/E is in NP.

Consider an arbitrary graph G and an integer k. We construct an instance (G, σ, k) of rsF-C/E by coloring all vertices distinctly. Then condition (ii) in Thm. 8.3 is always satisfied. To see this, we note that for $k > 1$ there are no edges between colors in the auxiliary graph $\mathcal{A}_{\digamma}(\sigma, \mathcal{I})$ such that their corresponding unique vertices are in distinct independent sets $I, I' \in \mathcal{I}$. The problem therefore reduces to completion/editing of (G, σ) to a complete multipartite graph, which is equivalent to a complementary deletion/editing of the complement of (G, k) to a disjoint union of cliques, i.e., a cluster graph. Both Cluster Deletion and Cluster Editing are NP-hard [275]. ☐

Although Cluster Completion is polynomial (it is solved by computing the transitive closure, i.e., by completing every connected component to a clique), rsF-D remains open: Consider a colored complete multipartite graph (G, σ) that is not an rs-Fitch graph. Then solving Cluster Completion on the complement returns (G, σ), which by construction is not a solution to rsF-D.

8.5.2 *Editing LDT Graphs to Fitch Graphs*

In Sec. 7.6, we briefly described how LDT graphs may be estimated. In contrast, it seems much more difficult to infer a Fitch graph $(\digamma, \sigma)$ directly from data. To our knowledge, no method for this purpose has been proposed in

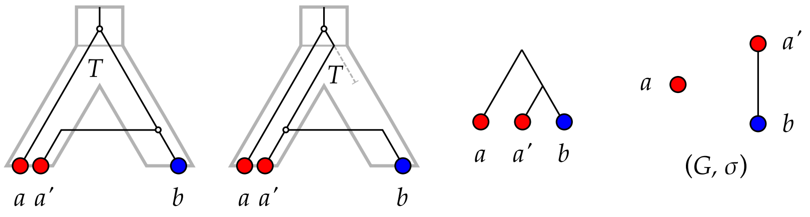


Fig. 73. Two relaxed scenarios with T displaying the triple $a'b|a$ and explaining the same graph (G,σ).

the literature. However, $(\digamma,\sigma)$ is of much more direct practical interest because the independent sets of $\digamma$ determine the maximal HGT-free subsets of genes, which could be analyzed separately by better-understood techniques. In this section, we therefore focus on the aspects of $(\digamma,\sigma)$ that are not captured by LDT graphs (G,σ). In the light of the previous sections, these are in particular non-replacing HGTs, i.e., HGTs that result in genes x and y in the same species $\sigma(x)=\sigma(y)$. In this case, $(\digamma,\sigma)$ is no longer properly colored and thus $G\neq\digamma$. To get a better intuition on this case consider three genes a, a', and b with $\sigma(a)=\sigma(a')\neq\sigma(b)$ with $ab\notin E(G)$ and $a'b\in E(G)$. By Lemma 7.7, the gene tree T of any explaining relaxed scenario displays the triple $a'b|a$. Fig. 73 shows two relaxed scenarios with a single HGT that explain this situation: In the first, we have $aa'\in E(\digamma)$, while the other implies $aa'\notin E(\digamma)$. Neither scenario is *a priori* less plausible than the other. Although the frequency of true homologous replacement via crossover decreases exponentially with the phylogenetic distance of donor and acceptor species [331], additive HGT with subsequent loss of one copy is an entirely plausible scenario.

A pragmatic approach to approximate $(\digamma,\sigma)$ is therefore to consider the step from an LDT graph (G,σ) to $(\digamma,\sigma)$ as a graph modification problem. First we note that Alg. 7 explicitly produces a relaxed scenario $\mathcal{S}$ and thus implies a corresponding gene tree $T_{\mathcal{S}}$ with HGT-labeling $\lambda_{\mathcal{S}}$, and thus an rs-Fitch graph $(\digamma(\mathcal{S}),\sigma)$. However, Alg. 7 was designed primarily as proof device. It produces neither a unique relaxed scenario nor necessarily the most plausible or a most parsimonious one. Furthermore, both the LDT graph (G,σ) and the desired rs-Fitch graph $(\digamma,\sigma)$ are consistent with a potentially very large number of relaxed scenarios. It thus appears preferable to altogether avoid the explicit construction of relaxed scenarios at this stage.

Since every LDT graph (G,σ) is explained by some $\mathcal{S}$, it is also a spanning subgraph of the corresponding rs-Fitch graph $(\digamma(\mathcal{S}),\sigma)$. The step from an LDT graph (G,σ) to an rs-Fitch graph $(\digamma,\sigma)$ can therefore be viewed as an edge-completion problem. The simplest variation of the problem is

Problem 8.2 (Fitch graph completion). *Given an LDT graph (G,σ), find a minimum cardinality set Q of possible edges such that $(G+Q,\sigma)$ is a complete multipartite graph.*

A close inspection of Problem 8.2 shows that the coloring is irrelevant in this version, and the actual problem to be solved is the problem Complete

Multipartite Graph Completion with a cograph as input. We next show that this task can be performed in linear time. The key idea is to consider the complementary problem, i.e., the problem of deleting a minimum set of edges from the complementary cograph $\overline{G}$ such that the end result is a disjoint union of complete graphs. This is known as Cluster Deletion problem [275], and is known to have a greedy solution for cographs [98].

Lemma 8.7. *There is a linear-time algorithm to solve Problem 8.2 for every cograph G.*

Proof. Instead of inserting in the cograph G the minimum number of edges necessary to reach a complete multipartite graph, we consider the equivalent problem of *deleting* a minimal set Q of edges from its complement $\overline{G}$, which is also a cograph, to obtain the complement of a complete multipartite graph, i.e., the disjoint union of complete graphs. This problem is known as the Cluster Deletion problem [275], which is known to have an polynomial-time solution for cographs [98]: A greedy maximum clique partition of G is obtained by recursively removing a maximum clique K from G, see also [77]. For cographs, the greedy maximum clique partitions are the solutions of the Cluster Deletion problem [98, Thm. 1]. The Maximum Clique problem on cographs can be solved in linear time using the co-tree of G [57], which can also be obtained in linear time [57]. ☐

An efficient algorithm to solve the Cluster Deletion problem for cographs can be devised by making use of the recursive construction of a cograph along its discriminating cotree (T, t), see Eq. (2). Denote by $\mathcal{P}(u)$ the optimal clique partition of the cograph implied by the subtree $T(u)$ of the discriminating cotree (T, t). We think of $\mathcal{P}(u) := [Q_1(u), Q_2(u), \dots]$ as an ordered list, such that $|Q_i(u)| \geq |Q_j(u)|$ if $i < j$. It will be convenient to assume that the list contains an arbitrary number of empty sets acting as an identity element for the join and disjoint union operation. With this convention, the optimal clique partitions $\mathcal{P}(u)$ satisfy the recursion

$$\mathcal{P}(u) = \begin{cases} \bigcup\limits_{v \in \mathsf{child}_T(u)} \mathcal{P}(v) & \text{if } t(u) = 0 \\ \left[\bigcup\limits_{v \in \mathsf{child}_T(u)} Q_i(v) \;\middle|\; i = 1, 2, \dots \right] & \text{if } t(u) = 1 \\ [\{u\}, \emptyset, \dots] & \text{if } u \text{ is a leaf} \end{cases} \tag{41}$$

for all $u \in V(T)$. In the first case, where $t(u) = 0$, we assume that the union operation to obtain $\mathcal{P}(u) = [Q_1(u), Q_2(u), \dots]$ maintains the property $|Q_i(u)| \geq |Q_j(u)|$ if $i < j$. In an implementation, this can e.g. be achieved using k-way merging where $k = |\mathsf{child}_T(u)|$.

To see that the recursion is correct, it suffices to recall that the greedy clique partition is optimal for cographs as input [98] and to observe the following simple properties of cliques in cographs [57]: (i) a largest clique in a disjoint union of graphs is also a largest clique in any of its components. The optimal clique partition of a disjoint union of graphs is, therefore, the union of the optimal clique partitions of the constituent connected components. (ii) For a join of two or more graphs G_i, each maximum size clique Q is the join of

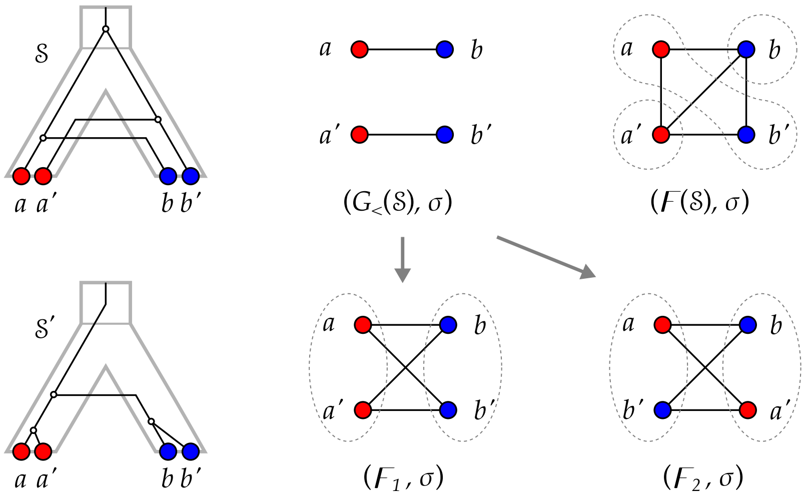


Fig. 74. Upper panel: a relaxed scenario $\mathcal{S}$ with LDT graph $(G_<(\mathcal{S}),\sigma)$ and rs-Fitch graph $(\digamma(\mathcal{S}),\sigma)$. There are two minimum edge completion sets that yield the complete multipartite graphs $(\digamma_1,\sigma)$ and $(\digamma_2,\sigma)$ (lower part). By Thm. 8.3, $(\digamma_2,\sigma)$ is not an rs-Fitch graph. The graph $(\digamma_1,\sigma)$ is an rs-Fitch graph for the relaxed scenario $\mathcal{S}'$. However, $G_<(\mathcal{S}) \neq G_<(\mathcal{S}')$ for all relaxed scenarios $\mathcal{S}'$ with $(\digamma(\mathcal{S}'),\sigma) = (\digamma_1,\sigma)$. To see this, note that the gene tree $T = ((a,b),(a',b'))$ in $\mathcal{S}$ is uniquely determined by application of Lemma 7.5 and 7.7. Assume that there is any edge-labeling λ such that $\digamma(T,\lambda) = \digamma_1$. The none-edges in $\digamma_1$ imply that along the two paths from a to a' and b to b' there is no transfer edge, that is, there cannot be any transfer edge in T; a contradiction.

a maximum size clique of each constituent. The next largest clique disjoint from $Q = \triangledown_i Q_i$ is, thus, the join of a largest cliques disjoint from Q_i in each constituent graph G_i. Thus a greedy clique partition of G is obtained by size ordering the clique partitions of G_i and joining the k-largest cliques from each.

The recursive construction of $\mathcal{P}(\rho_T)$ operates directly on the discriminating cotree (T,t) of the cograph G. For each node u, the effort is proportional to $|L(T(u))|\log(\deg(u))$ for the $\deg(u)$-wise merge sort step if $t(u) = 0$ and proportional to $|L(T(u))|$ for the merging of the k-th largest clusters for $t(u) = 1$. Using $\sum_u \deg(u)|L(T(u))| \leq |L(T)|\sum_u \deg(u) \leq |L(T)|2|E(T)|$ together with $|E(T)| = |V(T)| - 1$ and $|V(T)| \leq 2|L(T)| - 1$ (cf. [133, Lemma 1]), we obtain $\sum_u \deg(u)|L(T(u))| \in O(|L(T)|^2) = O(|V(G)|^2)$, that is, a quadratic upper bound on the running time.

In order to complete a given LDT graph (G,σ) to a Fitch graph, i.e., a complete multipartite graph, we can thus apply the recursive procedure in Eq. (41) to the discriminating cotree (T,t) of $\overline{G}$. This yields a partition $\mathcal{P}(\rho_T)$ of $V(G)$ which corresponds to the independent sets of the complete multipartite graph $\digamma$, and thus defines a solution $Q := E(\digamma) \setminus E(G)$ for Problem 8.2.

All maximum clique partitions of a cograph G have the same sequence of cluster sizes [98, Thm. 1]. However, they are not unique as partitions of the vertex set $V(G)$. Thus the minimal editing set Q that needs to be inserted into a cograph to reach a complete multipartite graphs will not be unique in general.

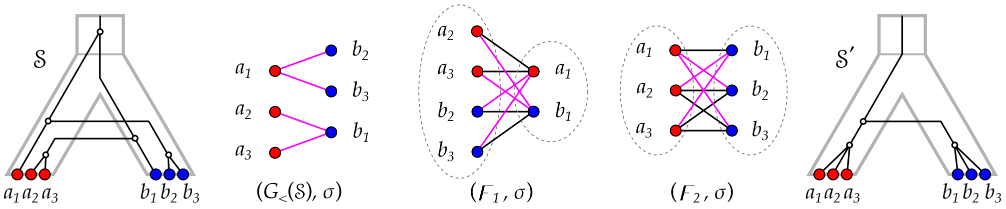


Fig. 75. The LDT graph $(G_<(\mathcal{S}),\sigma)$ for the relaxed scenario $\mathcal{S}$ has a unique minimum edge completion set (as determined by full enumeration), resulting in the complete multipartite graph (F_1,σ). However, Thm. 8.3 implies that (F_1,σ) is not rs-Fitch graph. An edge completion set with more edges must be used to obtain an rs-Fitch graph, for instance (F_2,σ), which is explained by the relaxed scenario $\mathcal{S}'$.

However, an optimal solution to Problem 8.2 with input (G,σ) does not necessarily yield an rs-Fitch graph or an rs-Fitch graph $(F(\mathcal{S}),\sigma)$ such that $G = G_<(\mathcal{S})$, see Fig. 74. In particular, there are LDT graphs (G,σ) for which more edges need to be added to obtain an rs-Fitch graph than the minimum required to obtain a complete multipartite graph, see Fig. 75.

A more relevant problems for our purposes, therefore is

Problem 8.3 (rs-Fitch graph completion). *Given an LDT graph* (G,σ) *find a minimum cardinality set* Q *of possible edges such that* $(G+Q,\sigma)$ *is an rs-Fitch graph.*

The following, stronger version is what we ideally would like to solve:

Problem 8.4 (strong rs-Fitch graph completion). *Given an LDT graph* (G,σ) *find a minimum cardinality set* Q *of possible edges such that* $(F := G+Q,\sigma)$ *is an rs-Fitch graph and there is a common scenario* $\mathcal{S}$*, i.e.,* $\mathcal{S}$ *satisfies* $G = G_<(\mathcal{S})$ *and* $F = F(\mathcal{S})$*.*

The computational complexity of Problems 8.3 and 8.4 is unknown. We conjecture, however, that the decision versions of both problems are NP-hard. In contrast to the application of graph modification problems to correct possible errors in the originally estimated data, the minimization of inserted edges into an LDT graph lacks a direct biological interpretation. Instead, most-parsimonious solutions in terms of evolutionary events are usually of interest in biology. In our framework, this translates to

Problem 8.5 (Min Transfer Completion). *Let* (G,σ) *be an LDT graph and* $\mathbb{S}$ *be the set of all relaxed scenarios* $\mathcal{S}$ *with* $G = G_<(\mathcal{S})$*. Find a relaxed scenario* $\mathcal{S}' \in \mathbb{S}$ *that has a minimal number of transfer edges among all elements in* $\mathbb{S}$ *and the corresponding rs-Fitch graph* $F(\mathcal{S}')$*.*

One way to address this problem might be as follows: Find edge-completion sets for the given LDT graph (G,σ) that minimize the number of independent sets in the resulting rs-Fitch graph $(F := G+Q,\sigma)$. The intuition behind this idea is that, in this case, the number of pairs within the individual independent sets is maximized and thus, we get a maximized set of gene pairs without transfer along their connecting path in the gene tree. It remains an open question whether this idea always yields a solution for Problem 8.5.

Evolutionary scenarios covering a wide range of HGT frequencies were generated with the simulation library `AsymmeTree` [293]. As described in Sec. 3.4, the tool generates a planted species tree S with time map τ_S. A constant-rate birth-death process then generates a "true" gene tree $(\widetilde{T}, \widetilde{\tau}_T)$, i.e., a tree that still contains leaves corresponding to losses, with additional branching events producing copies at inner vertex u of S propagating to each descendant lineage of u. To model HGT events, a recipient branch of S is selected at random. The simulation is event-based in the sense that each node of the true gene tree other than the planted root is one of speciation, gene duplication, horizontal gene transfer, gene loss, or a surviving gene. Here, the lost as well as the surviving genes form the leaf set of $\widetilde{T}$.

We used the following parameter settings for `AsymmeTree`: Planted species trees with a number of leaves between 10 and 50 (randomly drawn in each scenario) were generated using the Innovation Model [168] and equipped with a time map as described in [293]. Multifurcations were introduced into the species tree by contraction of inner edges with a common probability $p = 0.2$ per edge. Gene trees therefore are also not binary in general. We used multifurcations to model the effects of limited phylogenetic resolution. Duplication and HGT events, however, always result in bifurcations in the gene tree $\widetilde{T}$ (i.e., the parameter λ was set to zero). To avoid empty LDT and Fitch graphs, losses were constrained in a such way that at least one gene per gene tree survives. We considered different combinations of duplication, loss, and HGT event rates (indicated on the horizontal axis in Figs. 76–78). For each combination of event rates, we simulated 1000 scenarios per event rate combination. Fig. 76 summarizes basic statistics of the simulated data sets.

The simulation also determines the set of surviving genes $L \subseteq L(\widetilde{T})$, the reconciliation map $\widetilde{\mu}\colon V(\widetilde{T}) \to V(S) \cup E(S)$ and the coloring $\sigma\colon L \to L(S)$ representing the species in which each surviving gene resides. From the true tree $\widetilde{T}$, the observable gene tree $T = \widetilde{T}_{|L}$ is obtained by recursively removing leaves that correspond to loss events, i.e. $L(\widetilde{T}) \setminus L$, and suppressing inner vertices with a single child and setting $\tau_T(x) = \widetilde{\tau}_T(x)$ and $\mu(x) = \widetilde{\mu}(x)$ for all $x \in V(T)$. This defines a relaxed scenario $\mathscr{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$. From the scenario $\mathscr{S}$, we can immediately determine the associated HGT map $\lambda_{\mathscr{S}}$, the Fitch graph $\digamma(\mathscr{S})$, and the LDT graph $G_{<}(\mathscr{S})$. We also consider $\widetilde{\mathscr{S}} = (\widetilde{T}, S, \sigma, \widetilde{\mu}, \widetilde{\tau}_T, \tau_S)$ which, from a formal point of view, is not a relaxed scenario, see Fig. 77. In this example, the gene-species association $\sigma\colon L \to L(S)$ is not a map for the entire leaf set $L(\widetilde{T})$. Still, we can define the *true LDT graph* $G_{<}(\widetilde{\mathscr{S}})$ and the *true Fitch graph* $\digamma(\widetilde{\mathscr{S}})$ of $\widetilde{\mathscr{S}}$ in the same way as LDT graphs using Defs. 7.6, 7.7, and 8.3, respectively. Note that this does not guarantee that every true Fitch graph is also an rs-Fitch graph. The example in Fig. 77 shows, furthermore, that $\digamma(\widetilde{\mathscr{S}})[L] \neq \digamma(\mathscr{S})$ is possible. For the LDT graphs, on the other hand, we have $G_{<}(\mathscr{S}) = G_{<}(\widetilde{\mathscr{S}})$ because $\widetilde{\mathscr{S}}$ and $\mathscr{S}$ are based on the same time maps.

The distinction between the true graph $\digamma(\widetilde{\mathscr{S}})[L]$ and the rs-Fitch graph $\digamma(\mathscr{S})$ is closely related to the definition of transfer edges. So far, we only took into account transfer edges uv in the (observable) gene trees T, for

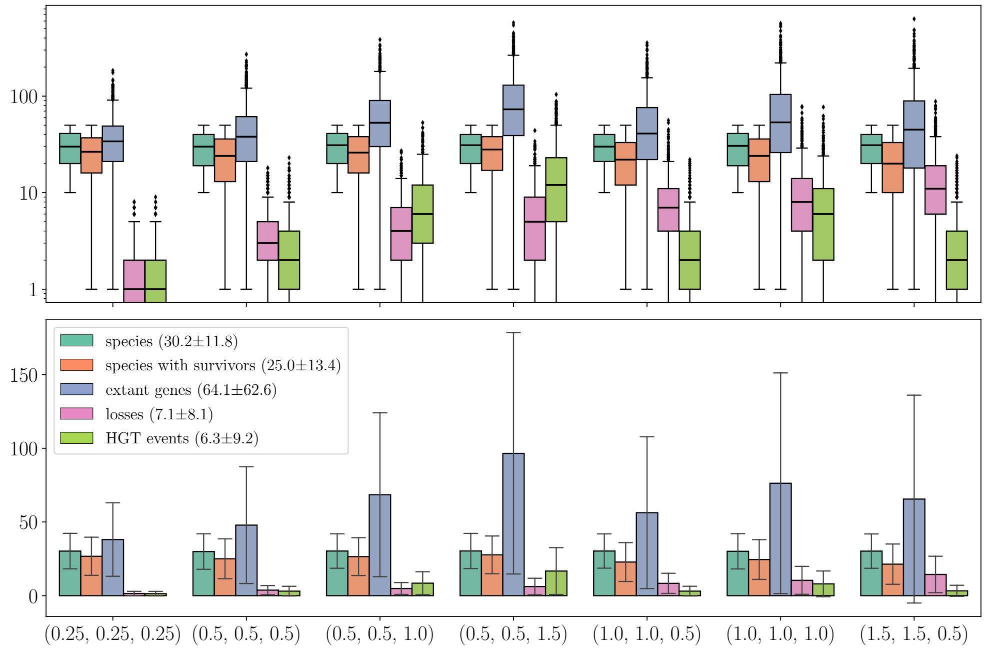


Fig. 76. Top panel: Distribution of the numbers of species (i.e. species tree leaves), species thereof that contain at least one surviving genes, surviving genes in total (non-loss leaves in the gene trees), loss events (loss leaves), and horizontal transfer events (inner vertices that are HGT events). Bottom panel: Mean and standard deviation of these quantities. The numbers in the legend indicate the mean and standard deviation taken over all event rate combinations. The tuples on the horizontal axis give the rates for duplication, loss, and horizontal transfer.

which u and v are mapped to incomparable vertices or edges of the species trees S (cf. Def. 8.1). Thus, given the knowledge of the relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$, these transfer edges are in that sense "visible". However, given $\widetilde{\mathcal{S}} = (\widetilde{T}, S, \sigma, \widetilde{\mu}, \widetilde{\tau}_T, \tau_S)$, which still contains all loss branches, it is possible that a non-transfer edge in T corresponds to a path in $\widetilde{T}$ which contains a transfer edge w.r.t. $\widetilde{\mathcal{S}}$, i.e., some edge $uv \in E(\widetilde{T})$ such that $\widetilde{\mu}(u)$ and $\widetilde{\mu}(v)$ are incomparable in S. In particular, this is the case whenever a gene is transferred into some recipient branch followed by a back-transfer into the original branch and a loss in the recipient branch (see Fig. 77, right). Fig. 77 shows that, in the majority of the simulated scenarios, the HGT information is preserved in the observable data. In fact, $\mathcal{F}(\mathcal{S}) = \mathcal{F}(\widetilde{\mathcal{S}})[L]$ in 86.7% of the simulated scenarios. Occasionally, however, we also encounter scenarios in which large fractions of the xenologous pairs are hidden from inference by the LDT-based approach.

Note that the same issue arises in the restriction of a scenario to a subset L' of leaves of T and to a subset M' of leaves of S, which is well-defined as long as $\sigma(L') \subseteq M'$. One can also define a corresponding restriction of the reconciliation map μ. Most importantly, the deletion of some leaves of T may leave inner vertices in T with only a single child, which are then suppressed to recover a phylogenetic tree. This replaces paths in T by single edges and

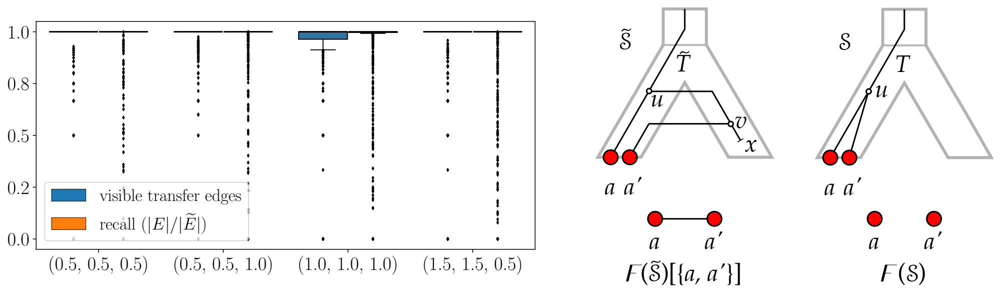


Fig. 77. Left: Fraction of "visible" transfer edges among the "true" transfer edges in T in the simulated scenarios, i.e., the edges that correspond to a path in $\widetilde{T}$ containing at least one transfer edge w.r.t. $\widetilde{\mathcal{S}}$ (see also the explanation in the text). The tuples on the horizontal axis give the rates for duplication, loss, and horizontal transfer. Since $E := E(\digamma(\mathcal{S})) \subseteq \widetilde{E} := E(\digamma(\widetilde{\mathcal{S}})[L(T)])$, we also show the ratio $|E|/|\widetilde{E}|$. Right: A relaxed scenario $\mathcal{S} = (T, S, \sigma, \mu, \tau_T, \tau_S)$ with an "invisible" transfer edge ua' (as determined by the knowledge of $\widetilde{\mathcal{S}} = (\widetilde{T}, S, \sigma, \widetilde{\mu}, \widetilde{\tau}_T, \tau_S)$). In this example we have $\digamma(\widetilde{\mathcal{S}})[L(T) = \{a, a'\}] \neq \digamma(\mathcal{S})$.

thus affects the definition of the HGT map $\lambda_{\mathcal{S}}$ since a path in T that contains two adjacent vertices u_1, u_2 with incomparable images $\mu(u_1)$ and $\mu(u_2)$ may be replaced by an edge with comparable end points in the restricted scenario $\mathcal{S}'$. This means that HGT events may become invisible, and thus $\digamma(\mathcal{S}')$ is not necessarily an *induced* subgraph of $\digamma(\mathcal{S})$, but a subgraph that may lack additional edges. Note that this is in contrast to the *assumptions* made in the analysis of (directed) Fitch graphs of 0/1-edge-labeled graphs [101, 131], where the information on horizontal transfers is inherited upon restriction of (T, λ).

In the following, we will only be concerned with estimating a Fitch graph $\digamma(\mathcal{S})$, i.e., the graph resulting from the "visible" transfer edges. These were edgeless in about 17.7% of the observable scenarios $\mathcal{S}$ (all parameter combinations taken into account). In these cases the LDT and thus also the inferred Fitch graphs are edgeless. These scenarios were excluded from further analysis.

We first ask how well the LDT graph $G_<(\mathcal{S})$ approximates the Fitch graph $\digamma(\mathcal{S})$. As shown in Fig. 78, the recall is limited. Over a broad range of parameters, the LDT graph contains about a third of the xenologous pairs. This begs the question whether the solution of the editing Problem 8.2, obtained using the exact recursive algorithm in Eq. (41), leads to a substantial improvement. We find that recall indeed increases substantially, at very moderate levels of false positives. The editing approach achieves a median precision of well above 90% in most cases and a median recall of at least 60%, it provides results that are at the very least encouraging. We find that minimal edge completion (Problem 8.2) already yields an rs-Fitch graph in the vast majority of cases (99.8%, scenarios of all parameter combinations taken into account), even if we restrict the color set to $M' := \sigma(L)$ (instead of $L(S)$) and thus force surjectivity of the coloring σ. We note that the original LDT graph and the minimal edge completion may not always be explained by a common relaxed scenario. This suggests that it will be worthwhile to consider the more diffi-

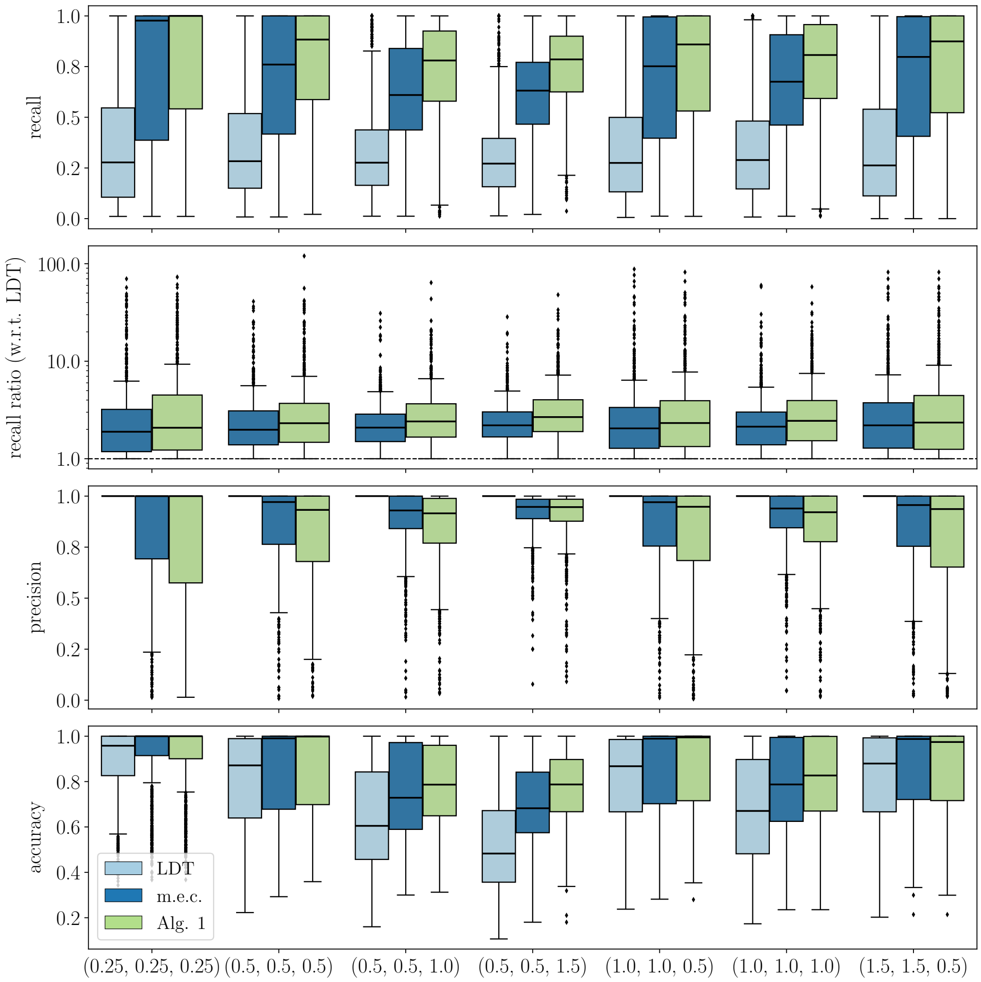


Fig. 78. Xenologs inferred from LDT graphs. Only observable scenarios $\mathcal{S}$ whose LDT graph $(G_<(\mathcal{S}), \sigma)$ contains at least one edge are included (82.3% of all scenarios). The tuples on the horizontal axis give the rates for duplication, loss, and horizontal transfer. Top panel: Recall. Fraction of edges in $\mathcal{F}(\mathcal{S})$ represented in $G_<(\mathcal{S})$ (light blue). As an alternative, the fraction of edges in a "minimum edge completion" (m.e.c.) to the "closest" complete multipartite graph is shown in dark blue. We observe a substantial increase in the fraction of inferred edges. The Fitch graph $\mathcal{F}(\mathcal{S}')$ obtained from the relaxed scenario $\mathcal{S}'$ produced by Alg. 7 with input $(G_<(\mathcal{S}), \sigma)$ yields an even better recall (light green). Second panel: Increase in the number of correctly inferred edges relative to the LDT graph $G_<(\mathcal{S})$. Third panel: Precision. In contrast to LDT graphs, which by Thm. 8.1 cannot contain false positive edges, this is not the case for the estimated Fitch graphs obtained as m.e.c. and by Alg. 7. While false positive edges are typically rare, occasionally very poor estimates are observed. Bottom panel: Accuracy.

cult editing problems for rs-Fitch graphs with a relaxed scenario $\mathscr{S}$ that at the same time explains the LDT graph.

Alg. 7 provides a means to obtain an rs-Fitch graph satisfying the latter constraint but without giving any guarantees for optimality in terms of a minimal edge completion. An implementation is available in the current release of the `AsymmeTree` package. For the rs-Fitch graphs $\digamma(\mathscr{S}')$ of the relaxed scenarios $\mathscr{S}'$ constructed by Alg. 7 with $(G_<(\mathscr{S}), \sigma)$ as input, we observe another moderate increase of recall when compared with the minimal edge completion results. This comes, however, at the expense of a loss in precision. This is not surprising, since $\digamma(\mathscr{S}')$ by construction contains at least as many edges as any minimal edge completion of $G_<(\mathscr{S})$. Therefore, the number of both true positive and false positive edges in $\digamma(\mathscr{S}')$ can be expected to be higher, resulting in a higher recall and lower precision, respectively.

The recall is given by $TP/(TP+FN)$, and $|E(\digamma(\mathscr{S}))| = TP+FN$ in terms of true positives TP and false negatives FN. Moreover, $G_<(\mathscr{S})$ is a subgraph of the Fitch graphs $\digamma_{\text{m.e.c.}}$ and $\digamma(\mathscr{S}')$ inferred with editing or with Alg. 7, respectively. The ratio $|E(\digamma(\mathscr{S})) \cap E(\digamma^*)| / |E(\digamma(\mathscr{S}) \cap E(G_<(\mathscr{S})))|$ with $\digamma^* \in \{\digamma_{\text{m.e.c.}}, \digamma(\mathscr{S}')\}$ therefore directly measures the increase in the number of correctly predicted xenologous pairs relative to the LDT. It is equivalent to the ratio of the respective recalls. By construction, the ratio is always ≥ 1. This is summarized as the second panel in Fig. 78.

8.7 SUMMARY

In this chapter, we have formalized HGT events in the context of relaxed scenarios by introducing HGT labelings (Def. 8.1) and rs-Fitch graphs (Def. 8.3). By Thm. 8.1, the LDT graph of a scenario is always a subgraph of its rs-Fitch graph. We further showed that LDT graphs describe a sufficient set of HGT events if and only if they are complete multipartite graphs (Thm. 8.2). This corresponds to scenarios in which all HGT events are replacing. Otherwise, additional HGT events exist that separate genes from the same species. To better understand these, we investigated scenario-derived rs-Fitch graphs and characterized them in Thm. 8.3 as those complete multipartite graphs that satisfy an additional constraint on the coloring (expressed in terms of an auxiliary graph). Although the information contained in LDT graphs is not sufficient to unambiguously determine the missing HGT edges, we arrive at an efficiently solvable graph editing problem from which a "best guess" can be obtained. To our knowledge, this is the first detailed mathematical investigation into the power and limitation of an implicit phylogenetic method for HGT inference.

We complement the theoretical findings with simulated scenarios, for which we find that LDT graphs cover roughly a third of xenologous pairs, while a simple greedy graph editing scheme can more than double the recall at moderate false positive rates. This greedy approach already yields a median accuracy of 89%, and in 99.8% of the cases produces biologically feasible solutions in the sense that the inferred graphs are rs-Fitch graphs.

9

CONCLUSION

Graphs appear naturally in computational biology as a means of representing answers to yes/no questions in pairwise comparisons of gene family members. Methods following this paradigm are often considered to be more robust than e.g. gene tree reconstruction methods which require that evolution approximately followed a well-behaved Markov process [15, 133, 293]. Moreover, under the assumption that genes evolve in a tree-like manner, the graphs resulting from certain yes/no questions often satisfy specific mathematical properties [101, 102, 132, 180]. This, in turn, provides an avenue for the correction of initial graph estimates from real-life data which, due to noise and measurement errors, cannot be expected to perfectly meet the constraints in general.

This work focused on *best match graphs (BMGs)* and *later-divergence-time (LDT) graphs*, as well as their close relationship with orthology and xenology, respectively, being two well-known modes of homology.

Best match graphs are centered around the question of whether a gene b in species B is at least as closely related to a gene a from species A as any other gene from B. BMGs and their symmetric parts, the RBMGs, have been introduced and studied earlier in [102, 104], and revisited in this work. This has led to a number of interesting novel insights. In particular, we derived a corrected and at the same time simplified characterization of BMGs in terms of informative triples and the `BUILD` algorithm. Taking into account the set of forbidden triples gives an alternative characterization that avoids the explicit construction of a tree, and, therefore, has proven useful for ILP formulations of BMG modification problems. Not surprisingly, all of these problems are NP-complete in general, even if the input digraph has only two colors or if we constrain the editing result to be explainable by a fully resolved tree. Such *binary-explainable BMGs (beBMGs)* have not received much attention before. Here, we characterized them in terms of a forbidden induced subgraph – the *hourglass* – and gave a near-cubic algorithm for the construction of their unique *binary-resolvable tree (BRT)*. Similarly, a fast algorithm for the recognition of 2-colored BMGs and the construction of their least resolved trees (LRT) was developed.

Orthology can be estimated without the necessity for a gene and species tree reconciliation using best matches as an intermediate step. Consequently, there are two main sources of error in such an inference pipeline. The first is the estimation of valid BMGs from sequence similarity or distance data. While some systematic errors can be largely avoided using outgroup genes [293], we addressed the question of how to deal with the residual amount of error by means of arc modification problems. To this end, we explored several heuristic approaches to BMG editing. Simulation results suggest that, for moderate levels of noise, the underlying tree structure is preserved in a perturbed BMG enabling correction heuristics to produce good approximations of the unknown "true" BMG. The second problem is the reconstruction of the orthology graph from a BMG. While it is not possible to logically guarantee that reciprocal best matches are indeed orthologs, we showed that, for duplication-loss scenarios, all unambiguous false positives can be identified in polynomial time. This extends previous findings [75, 103] and can be achieved with the *augmented tree* which is a special refinement of the LRT. However, even though one obtains a unique orthology graph in this way, the latter may not be biologically feasible in the sense of [130, 140], i.e., there is no pair of gene and species tree that admits a reconciliation explaining the orthology graph. In this case, there is clear evidence for additional false positives which cannot be unambiguously identified with the knowledge of the BMG alone.

The relationship of best matches and orthology is less clear in the presence of HGT and presumably does not admit any mathematical guarantees in this case. For the application of the results above, it is therefore of interest to identify maximal HGT-free groups of genes, i.e., such that there is no pair among them, whose connecting path in the gene tree contains a horizontal transfer. These groups are exactly the (maximal) independent sets of the so-called Fitch graphs, whose edges correspond to pairs of xenologous genes. Once the HGT-free groups have been identified, they can be analyzed separately in the simpler duplication-loss framework, see also [130].

To this end, we conducted an in-depth mathematical investigation of the combinatorial constraints arising in implicit phylogenetic methods for HGT inference. More precisely, we introduced the LDT graph which connects two genes with an edge precisely if they diverged later than the respective species in which they reside. In the very general framework of *relaxed scenarios*, LDT graphs can be characterized as those properly-colored cographs for which a certain set of species triples is consistent. Moreover, in accordance with the main idea of implicit HGT detection, the LDT graph is always a subgraph of the so-called *rs-Fitch graph* of a relaxed scenario, i.e., all edges in an LDT graph correspond to pairs of xenologs.

From a data analysis point of view, LDT graphs appear to be an attractive avenue to infer HGT in practice. While existing methods to estimate them from (dis)similarity data can certainly be improved, it is possible to use their cograph structure to correct the initial estimate in the same way as orthology data [133]. Although the LDT graph modification problems are NP-complete, it does not appear too difficult to modify efficient cograph editing heuristics [59, 136] to accommodate the additional coloring constraints.

Several alternative routes could then be followed to obtain Fitch graphs from LDT graphs. The most straightforward way is to elaborate on the editing problems briefly discussed in Sec. 8.5. A natural question arising in this context is whether there are non-LDT edges that are shared by all minimal completion sets and whether these "obligatory Fitch-edges" can be determined efficiently. A natural alternative is to modify Alg. 7 to incorporate some form of a cost function to favor the construction of biologically plausible scenarios.

In a very different approach, one might also consider using LDT graphs as constraints in probabilistic models to reconstruct scenarios, see e.g. [170, 283].

LDT graphs by themselves clearly do not contain sufficient information to completely determine a relaxed scenario. Additional information, e.g. a BMG will certainly be required. Another unresolved question is whether the resulting HGT-free subtrees can be combined into a complete scenario using only relational information such as best match data. One way to tackle this is to employ the techniques used by Lafond and Hellmuth [181] to characterize the conditions under which a fully event-labeled gene tree can be reconciled with unknown species trees. These not only resulted in a polynomial-time algorithm but also establish additional constraints on the HGT-free subtrees. An alternative, albeit mathematically less appealing, approach is to adapt classical phylogenetic methods to accommodate the HGT-free subtrees as constraints. We suspect that best match data can supply further, stringent constraints for this task.

Given that a BMG and a Fitch graph have been estimated independently, one could ask whether there is a common scenario or at least a common leaf-colored, 0/1-edge-labeled tree (T, λ, σ) that explains them. For binary trees, this problem has been solved in an upcoming contribution [138]. Therein, it is characterized when a tree T (or a refinement T^* thereof) and a partition $\mathcal{P}$ of its leaf set are compatible, i.e., whether there is a subset $H \subseteq E(T)$ (or $H \subseteq E(T^*)$) of edges whose removal in the tree exactly induces the partition $\mathcal{P}$ of the leaf set. In particular, this can be decided in linear time. Hence, one can construct the BRT (T, σ) of a beBMG $(\vec{G}, \sigma)$, and check whether some refinement T^* of T and the partition $\mathcal{P}$ given by the independent sets of a Fitch graph are compatible. Since every refinement (T^*, σ) of the BRT again explains $(\vec{G}, \sigma)$, a common gene tree (T^*, λ, σ) can be obtained by setting $\lambda(e) = 1$ if and only if $e \in H$. Intriguingly, the results in [138] also suggest that, in some cases, it is possible to obtain a "direction" of the HGT in the sense of directed Fitch graph [101, 131] given that a BMG (and thus an LRT or BRT) is supplied.

Alternative approaches to directed Fitch graph inference may exploit the fact that LDT graphs are not the only conceivable type of accessible xenology information. A large class of methods is designed to assess whether a single gene is *a xenolog*, i.e., whether there is evidence that it has been horizontally inserted into the genome under consideration. Parametric methods e.g. evaluate nucleotide composition patterns to detect foreign genes. In particular, the recipient species is clear in this case. Such predictions have already been combined with implicit phylogenetic evidence for HGT using Bayesian methods [254]. It remains an open question how this information can be utilized in conjunction with LDT graphs to provide not only additional constraints to infer rs-Fitch graphs but also the desired directional information.

The notion of *relaxed reconciliation maps* used here appears to be at least as general as alternatives that have been explored in the literature. It avoids the concurrent definition of event types and thus allows situations that may be excluded in a more restrictive setting. For example, relaxed scenarios may have two or more vertically inherited genes x and y in the same species with

$u := \text{lca}_T(x,y)$ mapping to a vertex of the species trees. In the usual interpretation, u corresponds to a speciation event (by virtue of $\mu(u) \in V^0(S)$); on the other hand, the descendants x and y constitute paralogs in most interpretations. Such scenarios are explicitly excluded e.g. in [293]. Lemma 7.3 suggests that relaxed scenarios are sufficiently flexible to make it possible to replace a scenario $\mathcal{S}$ that is "forbidden" in response to such inconsistent interpretations of events by an "allowed" scenario $\mathcal{S}'$ with the same leaf coloring σ such that $G_<(\mathcal{S}) = G_<(\mathcal{S}')$. Whether this is indeed true, or whether a more restrictive definition of reconciliation imposes additional constraints for LDT graphs will of course need to be checked in each case.

We have discussed horizontal transfers which are invisible due to losses or restrictions of scenarios to subsets of leaves in Sec. 8.6. This issue is a special case of the more general problem with the *observability* of events. Conceptually, we assume that evolution followed a *true scenario* comprising discrete events (speciations, duplications, horizontal transfer, gene losses, and possibly other events such as hybridization which were not considered here). In computer simulations, of course, we know this true scenario, as well as all event types. Gene loss not only renders some leaves invisible but also erases the evidence of all subtrees without surviving leaves. Removal of these vertices in general results in a non-phylogenetic gene tree that contains inner vertices with a single child. In the absence of horizontal transfer, this causes little problems and the *unobservable vertices* can simply be suppressed, see also [140]. The situation is more complicated with HGT. In [228], an HGT-vertex is deemed observable if it has both a horizontally and a vertically inherited descendant. This type of "vertex-centered" notion of xenology is explored further in [134]. In our present setting, the scenario retains an HGT edge by virtue of consecutive vertices in T with incomparable μ-images, irrespective of whether an HGT vertex is retained. We suspect that these different points of view can be unified only when gene losses are represented explicitly or when gene and species trees are not required to be phylogenetic (with single-child vertices implicating losses). Either extension of the theory, however, requires a more systematic understanding of which losses need to be represented and what evidence can be acquired to "observe" them.

A closely related issue concerns the notion of orthology in relaxed scenarios. Pragmatically, one would define two genes x and y to be *orthologs* if $\mu(\text{lca}_T(x,y)) \in V^0(S)$, i.e., if x and y are the product of a speciation event. Lemma 7.3 implies that there always is a scenario without any orthologs that explains a given LDT graph (G,σ). In particular, therefore, (G,σ) makes no implications on orthology. Conversely, however, orthology information is available and additional information on HGT might become available. In a situation akin to Fig. 73 (with the ancestral duplication moved down to the speciation), knowing that a and b are orthologs in the more restrictive sense that $\mu(\text{lca}_T(a,b)) = \text{lca}_S(\sigma(a),\sigma(b))$ excludes the r.h.s. scenario and implies that a' is the horizontally inherited child, and therefore also that a and a' are xenologs. This connection of orthology and xenology is also the subject of ongoing research.

In summary, we have obtained characterizations of BMGs, LDT graphs, and rs-Fitch graphs, and investigated their theoretical potential for the reconstruction of the orthology and xenology relation. Still, many open questions and avenues for future research remain. Even though the simulation results are promising, the applicability of the findings to real-life data has yet to be assessed. From a theoretical perspective, we have marked out novel possibilities for the reconstruction of gene family histories from relational data. Complementarily, we have found a number of limitations of the frameworks considered here. It will be interesting to further explore what kind of models are required to resolve the remaining ambiguities, whether this can be achieved on the basis of pairwise comparisons, and, most importantly, what needs to be measured to obtain the necessary information.

BIBLIOGRAPHY

[1] S. Aaronson. Quantum computing and hidden variables. *Physical Review A*, 71(3):032325, 2005. ISSN 1050-2947, 1094-1622. doi: 10.1103/PhysRevA.71.032325.

[2] S. Aaronson. BQP and the polynomial hierarchy. In *Proceedings of the 42nd ACM symposium on Theory of computing - STOC '10*, page 141. ACM Press, 2010. ISBN 978-1-4503-0050-6. doi: 10.1145/1806689.1806711.

[3] R. Abbott, D. Albach, S. Ansell, J. W. Arntzen, S. J. E. Baird, N. Bierne, J. Boughman, A. Brelsford, C. A. Buerkle, R. Buggs, R. K. Butlin, U. Dieckmann, F. Eroukhmanoff, A. Grill, S. H. Cahan, J. S. Hermansen, G. Hewitt, A. G. Hudson, C. Jiggins, J. Jones, B. Keller, T. Marczewski, J. Mallet, P. Martinez-Rodriguez, M. Möst, S. Mullen, R. Nichols, A. W. Nolte, C. Parisod, K. Pfennig, A. M. Rice, M. G. Ritchie, B. Seifert, C. M. Smadja, R. Stelkens, J. M. Szymura, R. Väinölä, J. B. W. Wolf, and D. Zinner. Hybridization and speciation. *Journal of Evolutionary Biology*, 26(2):229–246, 2013. ISSN 1010061X. doi: 10.1111/j.1420-9101.2012.02599.x.

[4] S. S. Abby, E. Tannier, M. Gouy, and V. Daubin. Lateral gene transfer as a support for the tree of life. *Proceedings of the National Academy of Sciences*, 109(13):4962–4967, 2012. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1116871109.

[5] G. Abrams and J. K. Sklar. The graph menagerie: Abstract algebra and the mad veterinarian. *Mathematics Magazine*, 83:168–179, 2010. ISSN 0025-570X, 1930-0980. doi: 10.4169/002557010X494814.

[6] R. Acuña, B. E. Padilla, C. P. Flórez-Ramos, J. D. Rubio, J. C. Herrera, P. Benavides, S.-J. Lee, T. H. Yeats, A. N. Egan, J. J. Doyle, and J. K. C. Rose. Adaptive horizontal transfer of a bacterial gene to an invasive insect pest of coffee. *Proceedings of the National Academy of Sciences of the United States of America*, 109(11):4197–4202, 2012. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1121190109.

[7] 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 Journal on Computing*, 10:405–421, 1981. ISSN 0097-5397, 1095-7111. doi: 10.1137/0210030.

[8] R. Albalat and C. Cañestro. Evolution by gene loss. *Nature Reviews Genetics*, 17(7):379–391, 2016. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg.2016.39.

[9] B. Alberts, A. Johnson, J. Lewis, D. Morgan, M. Raff, K. Roberts, and P. Walter. *Molecular biology of the cell*. Garland Science, Taylor and Francis Group, sixth edition edition, 2015. ISBN 978-0-8153-4432-2 978-0-8153-4464-3 978-0-8153-4524-4.

[10] K. Alix, P. R. Gérard, T. Schwarzacher, and J. S. P. Heslop-Harrison. Polyploidy and interspecific hybridization: partners for adaptation, speciation and evolution in plants. *Annals of Botany*, 120(2):183–194, 2017. ISSN 0305-7364, 1095-8290. doi: 10.1093/aob/mcx079.

[11] A. M. Altenhoff and C. Dessimoz. Inferring orthology and paralogy. In M. Anisimova, editor, *Evolutionary Genomics*, volume 855, pages 259–279. Humana Press, 2012. ISBN 978-1-61779-581-7 978-1-61779-582-4. doi: 10.1007/978-1-61779-582-4_9.

[12] A. M. Altenhoff, R. A. Studer, M. Robinson-Rechavi, and C. Dessimoz. Resolving the ortholog conjecture: orthologs tend to be weakly, but significantly, more similar in function than paralogs. *PLoS Computational Biology*, 8(5):e1002514, 2012. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1002514.

[13] A. M. Altenhoff, M. Gil, G. H. Gonnet, and C. Dessimoz. Inferring hierarchical orthologous groups from orthologous gene pairs. *PLoS ONE*, 8(1):e53786, 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0053786.

[14] A. M. Altenhoff, B. Boeckmann, S. Capella-Gutierrez, D. A. Dalquen, T. DeLuca, K. Forslund, J. Huerta-Cepas, B. Linard, C. Pereira, L. P. Pryszcz, F. Schreiber, A. Sousa da Silva, D. Szklarczyk, C.-M. Train, P. Bork, O. Lecompte, C. von Mering, I. Xenarios, K. Sjölander, L. Juhl Jensen, M. J. Martin, M. Muffato, Quest for Orthologs consortium, T. Gabaldón, S. E. Lewis, P. D. Thomas, E. Sonnhammer, and C. Dessimoz. Standardized benchmarking in the quest for orthologs. *Nature Methods*, 13:425–430, 2016. ISSN 1548-7091, 1548-7105. doi: 10.1038/nmeth.3830.

[15] A. M. Altenhoff, N. M. Glover, and C. Dessimoz. Inferring orthology and paralogy. In M. Anisimova, editor, *Evolutionary Genomics*, volume 1910, pages 149–175. Springer New York, 2019. ISBN 978-1-4939-9073-3 978-1-4939-9074-0. doi: 10.1007/978-1-4939-9074-0_5.

[16] A. M. Altenhoff, J. Garrayo-Ventas, S. Cosentino, D. Emms, N. M. Glover, A. Hernández-Plaza, Y. Nevers, V. Sundesha, D. Szklarczyk, J. M. Fernández, L. Codó, t. Q. for Orthologs Consortium, J. L. Gelpi, J. Huerta-Cepas, W. Iwasaki, S. Kelly, O. Lecompte, M. Muffato, M. J. Martin, S. Capella-Gutierrez, P. D. Thomas, E. Sonnhammer, and C. Dessimoz. The quest for orthologs benchmark service and consensus calls in 2020. *Nucleic Acids Research*, 48:W538–W545, 2020. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkaa308.

[17] S. Altschul. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997. ISSN 13624962. doi: 10.1093/nar/25.17.3389.

[18] M. Anisimova, editor. *Evolutionary Genomics*, volume 855 of *Methods in Molecular Biology*. Humana Press, 2012. ISBN 978-1-61779-581-7 978-1-61779-582-4. doi: 10.1007/978-1-61779-582-4.

[19] S. Arora and B. Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009. ISBN 978-0-521-42426-4. OCLC: ocn286431654.

[20] L. Arvestad, A.-C. Berglund, J. Lagergren, and B. Sennblad. Bayesian gene/species tree reconciliation and orthology analysis using MCMC. *Bioinformatics*, 19:i7–i15, 2003. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btg1000.

[21] R. Assis and D. Bachtrog. Neofunctionalization of young duplicate genes in drosophila. *Proceedings of the National Academy of Sciences of the United States of America*, 110(43):17409–17414, 2013. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1313759110.

[22] J. A. Bailey, Z. Gu, R. A. Clark, K. Reinert, R. V. Samonte, S. Schwartz, M. D. Adams, E. W. Myers, P. W. Li, and E. E. Eichler. Recent segmental duplications in the human genome. *Science*, 297(5583):1003–1007, 2002. ISSN 00368075, 10959203. doi: 10.1126/science.1072047.

[23] H.-J. Bandelt and A. Dress. Reconstructing the shape of a tree from observed dissimilarity data. *Advances in Applied Mathematics*, 7(3):309–343, 1986. ISSN 01968858. doi: 10.1016/0196-8858(86)90038-2.

[24] M. S. Bansal, E. J. Alm, and M. Kellis. Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. *Bioinformatics*, 28:i283–i291, 2012. ISSN 1460-2059, 1367-4803. doi: 10.1093/bioinformatics/bts225.

[25] J. Becq, C. Churlaud, and P. Deschavanne. A benchmark of parametric methods for horizontal transfers detection. *PLoS ONE*, 5:e9989, 2010. ISSN 1932-6203. doi: 10.1371/journal.pone.0009989.

[26] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. ISSN 0097-5397, 1095-7111. doi: 10.1137/S0097539796300933.

[27] D. A. Benson, M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers. GenBank. *Nucleic Acids Research*, 41:D36–42, 2013. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gks1195.

[28] S. D. Bentley and J. Parkhill. Comparative genomic structure of prokaryotes. *Annual Review of Genetics*, 38:771–792, 2004. ISSN 0066-4197, 1545-2948. doi: 10.1146/annurev.genet.38.072902.094318.

[29] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10): P10008, 2008. ISSN 1742-5468. doi: 10.1088/1742-5468/2008/10/P10008.

[30] S. Böcker and A. W. M. Dress. Recovering symbolically dated, rooted trees from symbolic ultrametrics. *Advances in Mathematics*, 138:105–125, 1998. ISSN 00018708. doi: 10.1006/aima.1998.1743.

[31] S. Boettcher and S. Mertens. Analysis of the Karmarkar-Karp differencing algorithm. *The European Physical Journal B*, 65:131–140, 2008. ISSN 1434-6028, 1434-6036. doi: 10.1140/epjb/e2008-00320-9.

[32] P. Bork, T. Dandekar, Y. Diaz-Lazcoz, F. Eisenhaber, M. Huynen, and Y. Yuan. Predicting function: from genes to genomes and back 1 1edited by p. e. wright. *Journal of Molecular Biology*, 283(4):707–725, 1998. ISSN 00222836. doi: 10.1006/jmbi.1998.2144.

[33] P. J. Bowler. *Evolution: the history of an idea*. University of California Press, 3rd ed., completely rev. and expanded edition, 2003. ISBN 978-0-520-23693-6.

[34] J. Brosius. Retroposons–seeds of evolution. *Science*, 251(4995):753, 1991. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1990437.

[35] D. Bryant. *Building Trees, Hunting for Trees, and Comparing Trees: Theory and Methods in Phylogenetic Analysis*. Dissertation, University of Canterbury, Canterbury, NZ, 1997.

[36] D. Bryant and V. Moulton. Neighbor-Net: An agglomerative method for the construction of phylogenetic networks. *Molecular Biology and Evolution*, 21(2):255–265, 2003. ISSN 0737-4038, 1537-1719. doi: 10.1093/molbev/msh018.

[37] D. Bryant and M. Steel. Extension operations on sets of leaf-labeled trees. *Advances in Applied Mathematics*, 16:425–453, 1995. ISSN 01968858. doi: 10.1006/aama.1995.1020.

[38] P. Buneman. The recovery of trees from measures of dissimilarity. In *Mathematics the the Archeological and Historical Sciences*, pages 387–395. Edinburgh University Press, 1971. ISBN 9780852242131.

[39] P. Burzyn, F. Bonomo, and G. Durán. NP-completeness results for edge modification problems. *Discrete Applied Mathematics*, 154:1824–1844, 2006. ISSN 0166218X. doi: 10.1016/j.dam.2006.03.031.

[40] R. K. Butlin, J. Galindo, and J. W. Grahame. Review. sympatric, parapatric or allopatric: the most important way to classify speciation? *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1506):2997–3007, 2008. ISSN 0962-8436, 1471-2970. doi: 10.1098/rstb.2008.0076.

[41] J. Byrka, S. Guillemot, and J. Jansson. New results on optimizing rooted triplets consistency. *Discrete Applied Mathematics*, 158:1136–1147, 2010. ISSN 0166218X. doi: 10.1016/j.dam.2010.03.004.

[42] K. P. Byrne and K. H. Wolfe. Consistent patterns of rate asymmetry and gene loss indicate widespread neofunctionalization of yeast genes after whole-genome duplication. *Genetics*, 175(3):1341–1350, 2007. ISSN 1943-2631. doi: 10.1534/genetics.106.066951.

[43] N. A. Campbell, L. A. Urry, M. L. Cain, S. A. Wasserman, P. V. Minorsky, and J. B. Reece. *Campbell Biology*. Pearson, 9. ed., global ed edition, 2011. ISBN 978-0-321-73975-9.

[44] J. A. Carlson, A. Jaffe, A. Wiles, C. M. Institute, and A. M. Society, editors. *The Millennium Prize problems*. American Mathematical Society; For The Clay Mathematics Institute, 2006. ISBN 978-0-8218-3679-8.

[45] L. L. Cavalli-Sforza and A. W. Edwards. Phylogenetic analysis. models and estimation procedures. *American Journal of Human Genetics*, 19(3):233–257, 1967. ISSN 0002-9297.

[46] W. C. Chang and O. Eulenstein. Reconciling gene trees with apparent polytomies. In D. Z. Chen and D. T. Lee, editors, *Computing and Combinatorics. COCOON 2006*, volume 4112 of *Lecture Notes in Computer Science*, pages 235–244, Berlin, Heidelberg, 2006. Springer. doi: 10.1007/11809678_26.

[47] W.-C. Chang, G. J. Burleigh, D. F. Fernández-Baca, and O. Eulenstein. An ILP solution for the gene duplication problem. *BMC Bioinformatics*, 12(S1):S14, 2011. ISSN 1471-2105. doi: 10.1186/1471-2105-12-S1-S14.

[48] M. A. Charleston. Jungles: a new solution to the host–parasite phylogeny reconciliation problem. *Mathematical Biosciences*, 149:191–223, 1998. ISSN 00255564. doi: 10.1016/S0025-5564(97)10012-8.

[49] M. A. Charleston and S. L. Perkins. Traversing the tangle: Algorithms and applications for cophylogenetic studies. *Journal of Biomedical Informatics*, 39:62–71, 2006. ISSN 15320464. doi: 10.1016/j.jbi.2005.08.006.

[50] M. Chatzou, C. Magis, J.-M. Chang, C. Kemena, G. Bussotti, I. Erb, and C. Notredame. Multiple sequence alignment modeling: methods and applications. *Briefings in Bioinformatics*, 17:1009–1023, 2016. ISSN 1467-5463, 1477-4054. doi: 10.1093/bib/bbv099.

[51] Z.-Z. Chen, F. Deng, and L. Wang. Simultaneous identification of duplications losses and lateral gene transfers. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9, 2012. ISSN 1545-5963. doi: 10.1109/TCBB.2012.79.

[52] S. C. Choi, M. D. Rasmussen, M. J. Hubisz, I. Gronau, M. J. Stanhope, and A. Siepel. Replacing and additive horizontal gene transfer in *Streptococcus*. *Molecular Biology and Evolution*, 29:3309–3320, 2012. ISSN 1537-1719, 0737-4038. doi: 10.1093/molbev/mss138.

[53] B. Chor and T. Tuller. Finding a maximum likelihood tree is hard. *Journal of the ACM*, 53:722–744, 2006. ISSN 0004-5411, 1557-735X. doi: 10.1145/1183907.1183909.

[54] P.-A. Christin, E. J. Edwards, G. Besnard, S. F. Boxall, R. Gregory, E. A. Kellogg, J. Hartwell, and C. P. Osborne. Adaptive evolution of c(4) photosynthesis through recurrent lateral gene transfer. *Current Biology*, 22(5):445–449, 2012. ISSN 09609822. doi: 10.1016/j.cub.2012.01.054.

[55] G. D. P. Clarke, R. G. Beiko, M. A. Ragan, and R. L. Charlebois. Inferring genome trees by using a filter to eliminate phylogenetically discordant sequences and a distance matrix based on mean normalized BLASTP scores. *Journal of Bacteriology*, 184:2072–2080, 2002. ISSN 0021-9193, 1098-5530. doi: 10.1128/JB.184.8.2072-2080.2002.

[56] H. Cohn, R. Pemantle, and J. G. Propp. Generating a random sink-free orientation in quadratic time. *The Electronic Journal of Combinatorics*, 9:R10, 2002. ISSN 1077-8926. doi: 10.37236/1627.

[57] D. G. Corneil, H. Lerchs, and L. S. Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3:163–174, 1981. ISSN 0166218X. doi: 10.1016/0166-218X(81)90013-5.

[58] D. G. Corneil, Y. Perl, and K. L. Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14:926–934, 1981. ISSN 0097-5397, 1095-7111. doi: 10.1137/0214065.

[59] C. Crespelle. Linear-time minimal cograph editing. `http://perso.ens-lyon.fr/christophe.crespelle/publications/SUB_minimal-cograph-editing.pdf`, 2019.

[60] D. A. Dalquen, M. Anisimova, G. H. Gonnet, and C. Dessimoz. ALF—a simulation framework for genome evolution. *Molecular Biology and Evolution*, 29(4):1115–1123, 2012. ISSN 1537-1719, 0737-4038. doi: 10.1093/molbev/msr268.

[61] C. A. Darby, M. Stolzer, P. J. Ropp, D. Barker, and D. Durand. Xenolog classification. *Bioinformatics*, 33:640–649, 2017. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btw686.

[62] S. Das, P. Ghosh, S. Ghosh, and S. Sen. Oriented bipartite graphs and the Goldbach graph. Technical Report math.CO/1611.10259v6, arXiv, 2020.

[63] V. Daubin and G. J. Szöllősi. Horizontal gene transfer and the history of life. *Cold Spring Harbor Perspectives in Biology*, 8(4):a018036, 2016. ISSN 1943-0264. doi: 10.1101/cshperspect.a018036.

[64] V. Daubin, E. Lerat, and G. Perrière. The source of laterally transferred genes in bacterial genomes. *Genome Biology*, 4(9):R57, 2003. ISSN 14656906. doi: 10.1186/gb-2003-4-9-r57.

[65] A. A. Davín, E. Tannier, T. A. Williams, B. Boussau, V. Daubin, and G. J. Szöllősi. Gene transfers can date the tree of life. *Nature Ecology & Evolution*, 2(5):904–909, 2018. ISSN 2397-334X. doi: 10.1038/s41559-018-0525-3.

[66] M. Dayhoff and R. Schwartz. A model for evolutionary change in proteins. In *Atlas of Protein Sequence and Structure*, pages 345–352. National Biomedical Research Foundation, 1978.

[67] M. C. H. Dekker. Reconstruction methods for derivation trees. Master's thesis, Vrije Universiteit, Amsterdam, NL, 1986.

[68] F. Delsuc, H. Brinkmann, and H. Philippe. Phylogenomics and the reconstruction of the tree of life. *Nature Reviews Genetics*, 6:361–375, 2005. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg1603.

[69] T. F. Deluca, I.-H. Wu, J. Pu, T. Monaghan, L. Peshkin, S. Singh, and D. P. Wall. Roundup: a multi-genome repository of orthologs and evolutionary distances. *Bioinformatics*, 22 (16):2044–2046, 2006. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btl286.

[70] C. Deng, C.-H. C. Cheng, H. Ye, X. He, and L. Chen. Evolution of an antifreeze protein by neofunctionalization under escape from adaptive conflict. *Proceedings of the National Academy of Sciences*, 107(50):21593–21598, 2010. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1007883107.

[71] Y. Deng and D. Fernández-Baca. Fast compatibility testing for rooted phylogenetic trees. *Algorithmica*, 80:2453–2477, 2018. ISSN 0178-4617, 1432-0541. doi: 10.1007/s00453-017-0330-4.

[72] D. L. Des Marais and M. D. Rausher. Escape from adaptive conflict after duplication in an anthocyanin pathway gene. *Nature*, 454(7205):762–765, 2008. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature07092.

[73] R. DeSalle, R. Absher, and G. Amato. Speciation and phylogenetic resolution. *Trends in Ecology & Evolution*, 9:297–298, 1994. ISSN 01695347. doi: 10.1016/0169-5347(94)90034-5.

[74] C. Dessimoz, G. Cannarozzi, M. Gil, D. Margadant, A. Roth, A. Schneider, and G. H. Gonnet. OMA, a comprehensive, automated project for the identification of orthologs from complete genome data: Introduction and first achievements. In A. McLysaght and D. H. Huson, editors, *Comparative Genomics*, volume 3678, pages 61–72. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-28932-6 978-3-540-31814-9. doi: 10.1007/11554714_6.

[75] C. Dessimoz, B. Boeckmann, A. C. J. Roth, and G. H. Gonnet. Detecting non-orthology in the COGs database and other approaches grouping orthologs using genome-specific best hits. *Nucleic Acids Research*, 34:3309–3316, 2006. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkl433.

[76] C. Dessimoz, D. Margadant, and G. H. Gonnet. DLIGHT – lateral gene transfer detection using pairwise evolutionary distances in a statistical framework. In *RECOMB 2008: Research in Computational Molecular Biology*, volume 4955, pages 315–330, Heidelberg, 2008. Springer. doi: 10.1007/978-3-540-78839-3_27.

[77] A. Dessmark, A. Lingas, E.-M. Lundell, M. Persson, and J. Jansson. On the approximability of maximum and minimum edge clique partition problems. *International Journal of Foundations of Computer Science*, 18:217–226, 2007. ISSN 0129-0541, 1793-6373. doi: 10.1142/S0129054107004656.

[78] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer Berlin Heidelberg, 2017. ISBN 978-3-662-53621-6 978-3-662-53622-3. doi: 10.1007/978-3-662-53622-3.

[79] R. Dondi, M. Lafond, and N. El-Mabrouk. Approximating the correction of weighted and unweighted orthology and paralogy relations. *Algorithms for Molecular Biology*, 12:4, 2017. ISSN 1748-7188. doi: 10.1186/s13015-017-0096-x.

[80] J.-P. Doyon, C. Chauve, and S. Hamel. Space of gene/species trees reconciliations and parsimonious models. *Journal of Computational Biology*, 16(10):1399–1418, 2009. ISSN 1066-5277, 1557-8666. doi: 10.1089/cmb.2009.0095.

[81] J.-P. Doyon, C. Scornavacca, K. Y. Gorbunov, G. J. Szöllősi, V. Ranwez, and V. Berry. An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers. In E. Tannier, editor, *Comparative Genomics. RECOMB-CG 2010*, volume 6398 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, 2010. Springer. doi: 10.1007/978-3-642-16181-0_9.

[82] J.-P. Doyon, V. Ranwez, V. Daubin, and V. Berry. Models, algorithms and programs for phylogeny reconciliation. *Briefings in Bioinformatics*, 12:392–400, 2011. ISSN 1467-5463, 1477-4054. doi: 10.1093/bib/bbr045.

[83] C. Dufraigne, B. Fertil, S. Lespinats, A. Giron, and P. Deschavanne. Detection and characterization of horizontal transfers in prokaryotes using genomic signature. *Nucleic Acids Research*, 33:e6, 2005. ISSN 1362-4962. doi: 10.1093/nar/gni004.

[84] R. C. Edgar. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797, 2004. ISSN 1362-4962. doi: 10.1093/nar/gkh340.

[85] E. S. El-Mallah and C. J. Colbourn. The complexity of some edge deletion problems. *IEEE Transactions on Circuits and Systems*, 35:354–362, 1988. ISSN 0098-4094, 1558-1276. doi: 10.1109/31.1748.

[86] I. Elias. Settling the intractability of multiple alignment. *Journal of Computational Biology*, 13:1323–1339, 2006. ISSN 1066-5277, 1557-8666. doi: 10.1089/cmb.2006.13.1323.

[87] O. Eulenstein. A linear time algorithm for tree mapping. Technical Report No. 1046, St, Arbeitspapiere der GMD, 1997.

[88] J. Felsenstein. *Inferring phylogenies*. Sinauer Associates, 2004. ISBN 978-0-87893-177-4.

[89] W. M. Fitch. Distinguishing homologous from analogous proteins. *Systematic Zoology*, 19:99–113, 1970. ISSN 00397989. doi: 10.2307/2412448.

[90] W. M. Fitch. Toward defining the course of evolution: Minimum change for a specific tree topology. *Systematic Zoology*, 20(4):406, 1971. ISSN 00397989. doi: 10.2307/2412116.

[91] W. M. Fitch. Homology: a personal view on some of the problems. *Trends in Genetics*, 16:227–231, 2000. ISSN 01689525. doi: 10.1016/S0168-9525(00)02005-9.

[92] R. D. Fleischmann, M. D. Adams, O. White, R. A. Clayton, E. F. Kirkness, A. R. Kerlavage, C. J. Bult, J. F. Tomb, B. A. Dougherty, and J. M. Merrick. Whole-genome random sequencing and assembly of haemophilus influenzae rd. *Science*, 269(5223):496–512, 1995. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.7542800.

[93] A. Force, M. Lynch, F. B. Pickett, A. Amores, Y. L. Yan, and J. Postlethwait. Preservation of duplicate genes by complementary, degenerative mutations. *Genetics*, 151(4):1531–1545, 1999. ISSN 0016-6731.

[94] P. Forster, L. Forster, C. Renfrew, and M. Forster. Phylogenetic network analysis of SARS-CoV-2 genomes. *Proceedings of the National Academy of Sciences of the United States of America*, 117(17):9241–9243, 2020. ISSN 1091-6490. doi: 10.1073/pnas.2004999117.

[95] T. Gabaldón and E. V. Koonin. Functional and evolutionary implications of gene orthology. *Nature Reviews Genetics*, 14:360–366, 2013. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg3456.

[96] T. Gabaldón, C. Dessimoz, J. Huxley-Jones, A. J. Vilella, E. L. Sonnhammer, and S. Lewis. Joining forces in the quest for orthologs. *Genome Biology*, 10(9):403, 2009. ISSN 1465-6906. doi: 10.1186/gb-2009-10-9-403.

[97] M. Y. Galperin, D. M. Kristensen, K. S. Makarova, Y. I. Wolf, and E. V. Koonin. Microbial genome analysis: the COG approach. *Briefings in Bioinformatics*, 20:1063–1070, 2019. ISSN 1467-5463, 1477-4054. doi: 10.1093/bib/bbx117.

[98] Y. Gao, D. R. Hare, and J. Nastos. The cluster deletion problem for cographs. *Discrete Mathematics*, 313(23):2763–2771, 2013. ISSN 0012365X. doi: 10.1016/j.disc.2013.08.017.

[99] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[100] L. Gasieniec, J. Jansson, A. Lingas, and A. Ostlin. On the complexity of constructing evolutionary trees. *Journal of Combinatorial Optimization*, 3:183–197, 1999. ISSN 13826905. doi: 10.1023/A:1009833626004.

[101] M. Geiß, J. Anders, P. F. Stadler, N. Wieseke, and M. Hellmuth. Reconstructing gene trees from Fitch's xenology relation. *Journal of Mathematical Biology*, 77:1459–1491, 2018. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-018-1260-8.

[102] M. Geiß, E. Chávez, M. González Laffitte, A. López Sánchez, B. M. R. Stadler, D. I. Valdivia, M. Hellmuth, M. Hernández Rosales, and P. F. Stadler. Best match graphs. *Journal of Mathematical Biology*, 78:2015–2057, 2019. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-019-01332-9.

[103] M. Geiß, M. E. González Laffitte, A. López Sánchez, D. I. Valdivia, M. Hellmuth, M. Hernández Rosales, and P. F. Stadler. Best match graphs and reconciliation of gene trees with species trees. *Journal of Mathematical Biology*, 80:1459–1495, 2020. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-020-01469-y.

[104] M. Geiß, P. F. Stadler, and M. Hellmuth. Reciprocal best match graphs. *Journal of Mathematical Biology*, 80:865–953, 2020. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-019-01444-2.

[105] M. B. Gerstein, C. Bruce, J. S. Rozowsky, D. Zheng, J. Du, J. O. Korbel, O. Emanuelsson, Z. D. Zhang, S. Weissman, and M. Snyder. What is a gene, post-ENCODE? history and updated definition. *Genome Research*, 17(6):669–681, 2007. ISSN 1088-9051. doi: 10.1101/gr.6339607.

[106] M. Goodman, J. Czelusniak, G. W. Moore, A. E. Romero-Herrera, and G. Matsuda. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Biology*, 28(2):132–163, 1979. ISSN 1063-5157, 1076-836X. doi: 10.1093/sysbio/28.2.132.

[107] K. Y. Gorbunov and V. A. Lyubetsky. Reconstructing the evolution of genes along the species tree. *Molecular Biology*, 43:881–893, 2009. ISSN 0026-8933, 1608-3245. doi: 10.1134/S0026893309050197.

[108] P. Górecki. H-trees: a model of evolutionary scenarios with horizontal gene transfer. *Fundamenta Informaticae*, 103:105–128, 2010. ISSN 01692968. doi: 10.3233/FI-2010-321.

[109] P. Górecki and J. Tiuryn. DLS-trees: A model of evolutionary scenarios. *Theoretical Computer Science*, 359:378–399, 2006. ISSN 03043975. doi: 10.1016/j.tcs.2006.05.019.

[110] P. Górecki and J. Tiuryn. Inferring evolutionary scenarios in the duplication, loss and horizontal gene transfer model. In R. L. Constable and A. Silva, editors, *Logic and Program Semantics*, volume 7230 of *Lecture Notes in Computer Science*, pages 83–105. Springer, Berlin, Heidelberg, 2012. doi: 10.1007/978-3-642-29485-3_7.

[111] B. E. Goulet, F. Roda, and R. Hopkins. Hybridization in plants: Old ideas, new techniques. *Plant Physiology*, 173(1):65–78, 2017. ISSN 0032-0889, 1532-2548. doi: 10.1104/pp.16.01340.

[112] L. A. Graham, J. Li, W. S. Davidson, and P. L. Davies. Smelt was the likely beneficiary of an antifreeze gene laterally transferred between fishes. *BMC Evolutionary Biology*, 12:

190, 2012. ISSN 1471-2148. doi: 10.1186/1471-2148-12-190.

[113] R. L. Graham and L. R. Foulds. Unlikelihood that minimal phylogenies for a realistic biological study can be constructed in reasonable computational time. *Mathematical Biosciences*, 60:133–142, 1982. ISSN 00255564. doi: 10.1016/0025-5564(82)90125-0.

[114] S. Grünewald, M. Steel, and M. S. Swenson. Closure operations in phylogenetics. *Mathematical Biosciences*, 208:521–537, 2007. ISSN 00255564. doi: 10.1016/j.mbs.2006.11.005.

[115] R. Guigó, I. Muchnik, and T. F. Smith. Reconstruction of ancient molecular phylogeny. *Molecular Phylogenetics and Evolution*, 6:189–213, 1996. ISSN 10557903. doi: 10.1006/mpev.1996.0071.

[116] S. Guillemot and M. Mnich. Kernel and fast algorithm for dense triplet inconsistency. *Theoretical Computer Science*, 494:134–143, 2013. ISSN 03043975. doi: 10.1016/j.tcs.2012.12.032.

[117] S. Guindon and O. Gascuel. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Systematic Biology*, 52(5):696–704, 2003. ISSN 1076-836X, 1063-5157. doi: 10.1080/10635150390235520.

[118] Gurobi Optimization, LLC. Gurobi Optimizer reference manual, 2021. URL `http://www.gurobi.com`.

[119] D. Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge UK, 1997. ISBN 978-0-521-58519-4. doi: 10.1017/CBO9780511574931.

[120] J. Hacker, G. Blum-Oehler, I. Mühldorfer, and H. Tschäpe. Pathogenicity islands of virulent bacteria: structure, function and impact on microbial evolution. *Molecular Microbiology*, 23(6):1089–1097, 1997. ISSN 0950-382X, 1365-2958. doi: 10.1046/j.1365-2958.1997.3101672.x.

[121] M. S. Hafner and S. A. Nadler. Phylogenetic trees support the coevolution of parasites and their hosts. *Nature*, 332(6161):258–259, 1988. ISSN 0028-0836, 1476-4687. doi: 10.1038/332258a0.

[122] O. Hagen and T. Stadler. TreeSimGM: Simulating phylogenetic trees under general Bellman-Harris models with lineage-specific shifts of speciation and extinction in R. *Methods in Ecology and Evolution*, 9:754–760, 2018. ISSN 2041-210X, 2041-210X. doi: 10.1111/2041-210X.12917.

[123] M. T. Hallett and J. Lagergren. Efficient algorithms for lateral gene transfer problems. In *RECOMB '01: Proceedings of the fifth annual international conference on Computational biology*, pages 149–156, New York, NY, 2001. Association for Computing Machinery. doi: 10.1145/369133.369188.

[124] K. Hanada, A. Tezuka, M. Nozawa, Y. Suzuki, S. Sugano, A. J. Nagano, M. Ito, and S.-I. Morinaga. Functional divergence of duplicate genes several million years after gene duplication in *arabidopsis*. *DNA Research*, 25:327–339, 2018. ISSN 1340-2838, 1756-1663. doi: 10.1093/dnares/dsy005.

[125] D. Hasić and E. Tannier. Gene tree reconciliation including transfers with replacement is NP-hard and FPT. *Journal of Combinatorial Optimization*, 38:502–544, 2019. ISSN 1382-6905, 1573-2886. doi: 10.1007/s10878-019-00396-z.

[126] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. ISSN 1464-3510, 0006-3444. doi: 10.1093/biomet/57.1.97.

[127] X. He and J. Zhang. Rapid subfunctionalization accompanied by prolonged and substantial neofunctionalization in duplicate gene evolution. *Genetics*, 169(2):1157–1164, 2005. ISSN 1943-2631. doi: 10.1534/genetics.104.037051.

[128] Y.-J. He, T. N. D. Huynh, J. Jansson, and W.-K. Sung. Inferring phylogenetic relationships avoiding forbidden rooted triplets. *Journal of Bioinformatics and Computational Biology*, 4: 59–74, 2006. ISSN 0219-7200, 1757-6334. doi: 10.1142/S0219720006001709.

[129] J.-H. Hehemann, G. Correc, T. Barbeyron, W. Helbert, M. Czjzek, and G. Michel. Transfer of carbohydrate-active enzymes from marine bacteria to japanese gut microbiota. *Nature*, 464(7290):908–912, 2010. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature08937.

[130] M. Hellmuth. Biologically feasible gene trees, reconciliation maps and informative triples. *Algorithms for Molecular Biology*, 12:23, 2017. ISSN 1748-7188. doi: 10.1186/s13015-017-0114-z.

[131] M. Hellmuth and C. R. Seemann. Alternative characterizations of Fitch's xenology relation. *Journal of Mathematical Biology*, 79:969–986, 2019. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-019-01384-x.

[132] M. Hellmuth, M. Hernandez-Rosales, K. T. Huber, V. Moulton, P. F. Stadler, and N. Wieseke. Orthology relations, symbolic ultrametrics, and cographs. *Journal of Mathematical Biology*, 66:399–420, 2013. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-012-0525-x.

[133] M. Hellmuth, N. Wieseke, M. Lechner, H.-P. Lenhof, M. Middendorf, and P. F. Stadler. Phylogenomics with paralogs. *Proceedings of the National Academy of Sciences*, 112:2058–2063, 2015. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1412770112.

[134] M. Hellmuth, P. F. Stadler, and N. Wieseke. The mathematics of xenology: Di-cographs, symbolic ultrametrics, 2-structures and tree-representable systems of binary relations. *Journal of Mathematical Biology*, 75:299–237, 2017. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-016-1084-3.

[135] M. Hellmuth, Y. Long, M. Geiß, and P. F. Stadler. A short note on undirected Fitch graphs. *The Art of Discrete and Applied Mathematics*, 1:P1.08, 2018. ISSN 2590-9770. doi: 10.26493/2590-9770.1245.98c.

[136] M. Hellmuth, A. Fritz, N. Wieseke, and P. F. Stadler. Techniques for the cograph editing problem: Module merge is equivalent to edit P_4's. *The Art of Discrete and Applied Mathematics*, 3:#P2.01, 2020. ISSN 2590-9770. doi: 10.26493/2590-9770.1252.e71.

[137] M. Hellmuth, M. Geiß, and P. F. Stadler. Complexity of modification problems for reciprocal best match graphs. *Theoretical Computer Science*, 809:384–393, 2020. ISSN 03043975. doi: 10.1016/j.tcs.2019.12.033.

[138] M. Hellmuth, D. Schaller, and P. F. Stadler. Compatibility of partitions, hierarchies, and split systems. 2021. submitted; arxiv: 2104.14146.

[139] M. R. Henzinger, V. King, and T. Warnow. Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology. *Algorithmica*, 24:1–13, 1999. ISSN 0178-4617. doi: 10.1007/PL00009268.

[140] M. Hernandez-Rosales, M. Hellmuth, N. Wieseke, K. T. Huber, V. Moulton, and P. F. 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.

[141] J. E. Hernández-Salmerón and J. E. Moreno-Hagelsieb. Progress in quickly finding orthologs as reciprocal best hits: comparing blast, last, diamond and MMseqs2. *BMC Genomics*, 21:741, 2020. ISSN 1471-2164. doi: 10.1186/s12864-020-07132-6.

[142] C. E. Hinchliff, S. A. Smith, J. F. Allman, J. G. Burleigh, R. Chaudhary, L. M. Coghill, K. A. Crandall, J. Deng, B. T. Drew, R. Gazis, K. Gude, D. S. Hibbett, L. A. Katz, H. D. Laughinghouse, E. J. McTavish, P. E. Midford, C. L. Owen, R. H. Ree, J. A. Rees, D. E. Soltis, T. Williams, and K. A. Cranston. Synthesis of phylogeny and taxonomy into a comprehensive tree of life. *Proceedings of the National Academy of Sciences of the United States of America*, 112(41):12764–12769, 2015. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1423041112.

[143] K. Hiramatsu, L. Cui, M. Kuroda, and T. Ito. The emergence and evolution of methicillin-resistant staphylococcus aureus. *Trends in Microbiology*, 9(10):486–493, 2001. ISSN 0966842X. doi: 10.1016/s0966-842x(01)02175-8.

[144] G. A. Hoelzer and D. J. Meinick. Patterns of speciation and limits to phylogenetic resolution. *Trends in Ecology & Evolution*, 9:104–107, 1994. ISSN 01695347. doi: 10.1016/

0169-5347(94)90207-0.

[145] P. W. H. Holland. Evolution of homeobox genes. *Wiley Interdisciplinary Reviews: Developmental Biology*, 2(1):31–45, 2013. ISSN 17597684. doi: 10.1002/wdev.78.

[146] J. Holm, K. de Lichtenberg, and M. Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *Journal of the ACM*, 48:723–760, 2001. ISSN 0004-5411, 1557-735X. doi: 10.1145/502090.502095.

[147] J. P. Huelsenbeck and F. Ronquist. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, 2001. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/17.8.754.

[148] J. P. Huelsenbeck, B. Larget, R. E. Miller, and F. Ronquist. Potential applications and pitfalls of bayesian inference of phylogeny. *Systematic Biology*, 51(5):673–688, 2002. ISSN 1076-836X, 1063-5157. doi: 10.1080/10635150290102366.

[149] J. Huerta-Cepas, A. Bueno, J. Dopazo, and T. Gabaldón. PhylomeDB: a database for genome-wide collections of gene phylogenies. *Nucleic Acids Research*, 36:D491–496, 2008. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkm899.

[150] J. Huerta-Cepas, D. Szklarczyk, D. Heller, A. Hernández-Plaza, S. K. Forslund, H. Cook, D. R. Mende, I. Letunic, T. Rattei, L. Jensen, C. von Mering, and P. Bork. eggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses. *Nucleic Acids Research*, 47:D309–D314, 2018. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gky1085.

[151] A. L. Hughes. The evolution of functionally novel proteins after gene duplication. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 256(1346):119–124, 1994. ISSN 0962-8452, 1471-2954. doi: 10.1098/rspb.1994.0058.

[152] F. Husnik and J. P. McCutcheon. Functional horizontal gene transfer from bacteria to eukaryotes. *Nature Reviews Microbiology*, 16:67–79, 2018. ISSN 1740-1526, 1740-1534. doi: 10.1038/nrmicro.2017.137.

[153] D. H. Huson, R. Rupp, and C. Scornavacca. *Phylogenetic networks: concepts, algorithms and applications*. Cambridge University Press, 2010. ISBN 978-0-521-75596-2.

[154] IBM. ILOG CPLEX Optimization Studio 12.10.0, 2019. URL `https://www.ibm.com/docs/en/icos/12.10.0`.

[155] H. Innan and F. Kondrashov. The evolution of gene duplications: classifying and distinguishing between models. *Nature Reviews Genetics*, 11(2):97–108, 2010. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg2689.

[156] International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945, 2004. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature03001.

[157] J. Jansson. On the complexity of inferring rooted evolutionary trees. *Electronic Notes in Discrete Mathematics*, 7:50–53, 2001. ISSN 15710653. doi: 10.1016/S1571-0653(04)00222-7.

[158] J. Jansson, J. H. Ng, K. Sadakane, and W.-K. Sung. Rooted maximum agreement supertrees. *Algorithmica*, 43:293–307, 2005. ISSN 0178-4617, 1432-0541. doi: 10.1007/s00453-004-1147-5.

[159] J. Jansson, R. Lemence, and A. Lingas. The complexity of inferring a minimally resolved phylogenetic supertree. *SIAM Journal on Computing*, 41:272–291, 2012. ISSN 0097-5397, 1095-7111. doi: 10.1137/100811489.

[160] D. T. Jones, W. R. Taylor, and J. M. Thornton. The rapid generation of mutation data matrices from protein sequences. *Bioinformatics*, 8(3):275–282, 1992. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/8.3.275.

[161] T. H. Jukes and C. R. Cantor. Evolution of protein molecules. In *Mammalian Protein Metabolism*, pages 21–132. Elsevier, 1969. ISBN 978-1-4832-3211-9. doi: 10.1016/B978-1-4832-3211-9.50009-7.

[162] A. Kanhere and M. Vingron. Horizontal gene transfers in prokaryotes show differential preferences for metabolic and translational genes. *BMC Evolutionary Biology*, 9:9, 2009. ISSN 1471-2148. doi: 10.1186/1471-2148-9-9.

[163] D. R. Karger. Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm. In *SODA '93. Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 21–30, Philadelphia, PA, 1993. Society for Industrial and Applied Mathematics. doi: 10.5555/313559.313605.

[164] N. Karmarkar and R. M. Karp. The differencing method of set partitioning. Technical Report UCB/CSD-83-113, EECS Department, University of California, Berkeley, 1983.

[165] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, editors, *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations*, pages 85–103. Springer, Boston, MA, 1972. doi: 10.1007/978-1-4684-2001-2_9.

[166] K. Katoh, K. Misawa, K.-i. Kuma, and T. Miyata. MAFFT: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic Acids Research*, 30 (14):3059–3066, 2002. ISSN 13624962. doi: 10.1093/nar/gkf436.

[167] P. J. Keeling and J. D. Palmer. Horizontal gene transfer in eukaryotic evolution. *Nature Reviews Genetics*, 9:605–618, 2008. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg2386.

[168] S. Keller-Schmidt and K. Klemm. A model of macroevolution as a branching process based on innovations. *Advances in Complex Systems*, 15:1250043, 2012. ISSN 0219-5259, 1793-6802. doi: 10.1142/S0219525912500439.

[169] D. G. Kendall. On the generalized birth-and-death process. *The Annals of Mathematical Statistics*, 19:1–15, 1948. ISSN 0003-4851. doi: 10.1214/aoms/1177730285.

[170] M. A. Khan, O. Mahmudi, I. Ullah, L. Arvestad, and J. Lagergren. Probabilistic inference of lateral gene transfer events. *BMC Bioinformatics*, 17:431, 2016. ISSN 1471-2105. doi: 10.1186/s12859-016-1268-2.

[171] T. Kinene, J. Wainaina, S. Maina, and L. Boykin. Methods for rooting trees. In *Encyclopedia of Evolutionary Biology*, pages 489–493. Elsevier, 2016. ISBN 978-0-12-800426-5. doi: 10.1016/B978-0-12-800049-6.00215-8.

[172] R. M. Kliman, P. Andolfatto, J. A. Coyne, F. Depaulis, M. Kreitman, A. J. Berry, J. McCarter, J. Wakeley, and J. Hey. The population genetics of the origin and divergence of the *Drosophila simulans* complex species. *Genetics*, 156:1913–1931, 2000.

[173] E. V. Koonin. Orthologs, paralogs, and evolutionary genomics. *Annual Review of Genetics*, 39:309–338, 2005. ISSN 0066-4197, 1545-2948. doi: 10.1146/annurev.genet.39.073003.114725.

[174] A. Korchmaros. The structure of 2-colored best match graphs. Technical Report math.CO/2009.00447v2, arXiv, 2020.

[175] A. Korchmaros. Circles and paths in 2-colored best match graphs. Technical Report math.CO/2006.04100v1, arXiv, 2020.

[176] M. Kordi, S. Kundu, and M. S. Bansal. On inferring additive and replacing horizontal gene transfers through phylogenetic reconciliation. preprint, Bioinformatics, 2020.

[177] R. Korona. Gene dispensability. *Current Opinion in Biotechnology*, 22(4):547–551, 2011. ISSN 09581669. doi: 10.1016/j.copbio.2011.04.017.

[178] D. M. Kristensen, Y. I. Wolf, A. R. Mushegian, and E. V. Koonin. Computational methods for gene orthology inference. *Briefings in Bioinformatics*, 12(5):379–391, 2011. ISSN 1467-5463, 1477-4054. doi: 10.1093/bib/bbr030.

[179] E. V. Kriventseva, F. Tegenfeldt, T. J. Petty, R. M. Waterhouse, F. A. Simão, I. A. Pozdnyakov, P. Ioannidis, and E. M. Zdobnov. OrthoDB v8: update of the hierarchical catalog of orthologs and the underlying free software. *Nucleic Acids Research*, 43:D250–256, 2015. ISSN 1362-4962, 0305-1048. doi: 10.1093/nar/gku1220.

[180] M. Lafond and N. El-Mabrouk. Orthology and paralogy constraints: satisfiability and consistency. *BMC Genomics*, 15:S12, 2014. ISSN 1471-2164. doi: 10.1186/1471-2164-15-S6-S12.

[181] M. Lafond and M. Hellmuth. Reconstruction of time-consistent species trees. *Algorithms for Molecular Biology*, 15:16, 2020. ISSN 1748-7188. doi: 10.1186/s13015-020-00175-0.

[182] M. Lafond, M. Semeria, K. M. Swenson, E. Tannier, and N. El-Mabrouk. Gene tree correction guided by orthology. *BMC Bioinformatics*, 14(S15):S5, 2013. ISSN 1471-2105. doi: 10.1186/1471-2105-14-S15-S5.

[183] M. Lafond, C. Chauve, R. Dondi, and N. El-Mabrouk. Polytomy refinement for the correction of dubious duplications in gene trees. *Bioinformatics*, 30:i519–i526, 2014. ISSN 1460-2059, 1367-4803. doi: 10.1093/bioinformatics/btu463.

[184] M. Lafond, R. D. Dondi, and N. El-Mabrouk. The link between orthology relations and gene trees: A correction perspective. *Algorithms for Molecular Biology*, 11:4, 2016. ISSN 1748-7188. doi: 10.1186/s13015-016-0067-7.

[185] M. Lafond, M. Meghdari Miardan, and D. Sankoff. Accurate prediction of orthologs in the presence of divergence after duplication. *Bioinformatics*, 34(13):i366–i375, 2018. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/bty242.

[186] J. G. Lawrence and D. L. Hartl. Inference of horizontal genetic transfer from molecular data: an approach using the bootstrap. *Genetics*, 131:753–760, 1992. ISSN 0016-6731.

[187] J. G. Lawrence and H. Ochman. Amelioration of bacterial genomes: rates of change and exchange. *Journal of Molecular Evolution*, 44(4):383–397, 1997. ISSN 0022-2844. doi: 10.1007/pl00006158.

[188] J. G. Lawrence and H. Ochman. Molecular archaeology of the escherichia coli genome. *Proceedings of the National Academy of Sciences of the United States of America*, 95(16):9413–9417, 1998. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.95.16.9413.

[189] M. Lechner, S. Findeiß, L. Steiner, M. Marz, P. F. Stadler, and S. J. Prohaska. `Proteinortho`: detection of (co-)orthologs in large-scale analysis. *BMC Bioinformatics*, 12:124, 2011. ISSN 1471-2105. doi: 10.1186/1471-2105-12-124.

[190] M. Lechner, M. Hernandez-Rosales, D. Doerr, N. Wieseke, A. Thévenin, J. Stoye, R. K. Hartmann, S. J. Prohaska, and P. F. Stadler. Orthology detection combining clustering and synteny for very large datasets. *PLoS ONE*, 9:e105015, 2014. ISSN 1932-6203. doi: 10.1371/journal.pone.0105015.

[191] N. A. Lerminiaux and A. D. S. Cameron. Horizontal transfer of antibiotic resistance genes in clinical environments. *Canadian Journal of Microbiology*, 65(1):34–44, 2019. ISSN 0008-4166, 1480-3275. doi: 10.1139/cjm-2018-0275.

[192] F.-W. Li, J. C. Villarreal, S. Kelly, C. J. Rothfels, M. Melkonian, E. Frangedakis, M. Ruhsam, E. M. Sigel, J. P. Der, J. Pittermann, D. O. Burge, L. Pokorny, A. Larsson, T. Chen, S. Weststrand, P. Thomas, E. Carpenter, Y. Zhang, Z. Tian, L. Chen, Z. Yan, Y. Zhu, X. Sun, J. Wang, D. W. Stevenson, B. J. Crandall-Stotler, A. J. Shaw, M. K. Deyholos, D. E. Soltis, S. W. Graham, M. D. Windham, J. A. Langdale, G. K.-S. Wong, S. Mathews, and K. M. Pryer. Horizontal transfer of an adaptive chimeric photoreceptor from bryophytes to ferns. *Proceedings of the National Academy of Sciences of the United States of America*, 111 (18):6672–6677, 2014. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1319929111.

[193] L. Li, C. J. Stoeckert, and D. S. Roos. OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Research*, 13(9):2178–2189, 2003. ISSN 1088-9051. doi: 10.1101/gr.1224503.

[194] D. Liao. Concerted evolution: Molecular mechanisms and biological implications. *The American Journal of Human Genetics*, 64:24–30, 1999. ISSN 00029297. doi: 10.1086/302221.

[195] B. Linard, J. D. Thompson, O. Poch, and O. Lecompte. OrthoInspector: comprehensive orthology analysis and visual exploration. *BMC Bioinformatics*, 12:11, 2011. ISSN 1471-2105. doi: 10.1186/1471-2105-12-11.

[196] L. Liu, Z. Xi, S. Wu, C. C. Davis, and S. V. Edwards. Estimating phylogenetic trees from genome-scale data. *Annals of the New York Academy of Sciences*, 1360:36–53, 2015. ISSN 00778923. doi: 10.1111/nyas.12747.

[197] Y. Liu, J. Wang, J. Guo, and J. Chen. Complexity and parameterized algorithms for cograph editing. *Theoretical Computer Science*, 461:45–54, 2012. ISSN 03043975. doi: 10.1016/j.tcs.2011.11.040.

[198] L. Lovász. Coverings and colorings of hypergraphs. In *Proc. 4th Southeastern Conference of Combinatorics, Graph Theory, and Computing*, volume 8 of *Congr. Num.*, pages 3–12, Winnipeg, CA, 1973. Utilitas Mathematica Publishing.

[199] M. Lynch and J. S. Conery. The evolutionary fate and consequences of duplicate genes. *Science*, 290(5494):1151–1155, 2000. ISSN 00368075, 10959203. doi: 10.1126/science.290.5494.1151.

[200] W. Ma, D. Smirnov, J. Forman, A. Schweickart, C. Slocum, S. Srinivasan, and R. Libeskind-Hadas. DTL-RnB: Algorithms and tools for summarizing the space of DTL reconciliations. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15:411–421, 2018. ISSN 1545-5963. doi: 10.1109/TCBB.2016.2537319.

[201] W. Maddison. Reconstructing character evolution on polytomous cladograms. *Cladistics*, 5:365–377, 1989. ISSN 07483007, 10960031. doi: 10.1111/j.1096-0031.1989.tb00569.x.

[202] W. P. Maddison and L. L. Knowles. Inferring phylogeny despite incomplete lineage sorting. *Systematic Biology*, 55(1):21–30, 2006. ISSN 1076-836X, 1063-5157. doi: 10.1080/10635150500354928.

[203] S. Magadum, U. Banerjee, P. Murugan, D. Gangapur, and R. Ravikesavan. Gene duplication as a major force in evolution. *Journal of Genetics*, 92(1):155–161, 2013. ISSN 0022-1333, 0973-7731. doi: 10.1007/s12041-013-0212-8.

[204] T. Mailund, G. S. Brodal, R. Fagerberg, C. N. S. Pedersen, and D. Phillips. Recrafting the neighbor-joining method. *BMC Bioinformatics*, 7:29, 2006. ISSN 14712105. doi: 10.1186/1471-2105-7-29.

[205] T. Mailund, K. Munch, and M. H. Schierup. Lineage sorting in apes. *Annual Review of Genetics*, 48:519–535, 2014. ISSN 0066-4197, 1545-2948. doi: 10.1146/annurev-genet-120213-092532.

[206] J. Mallet. Hybridization, ecological races and the nature of species: empirical evidence for the ease of speciation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1506):2971–2986, 2008. ISSN 0962-8436, 1471-2970. doi: 10.1098/rstb.2008.0081.

[207] D. Mallo, L. De Oliveira Martins, and D. Posada. SimPhy: Phylogenomic simulation of gene, locus, and species trees. *Systematic Biology*, 65(2):334–344, 2016. ISSN 1063-5157, 1076-836X. doi: 10.1093/sysbio/syv082.

[208] E. Mayr. *Systematics and the Origin of Species*. Columbia Univ. Press, 1942.

[209] R. S. McInnes, G. E. McCallum, L. E. Lamberte, and W. van Schaik. Horizontal transfer of antibiotic resistance genes in the human gut microbiome. *Current Opinion in Microbiology*, 53:35–43, 2020. ISSN 1879-0364. doi: 10.1016/j.mib.2020.02.002.

[210] T. A. McKee and F. R. McMorris. *Topics in Intersection Graph Theory*. Society for Industrial and Applied Mathematics, 1999. doi: 10.1137/1.9780898719802.

[211] R. W. Meredith, G. Zhang, M. T. P. Gilbert, E. D. Jarvis, and M. S. Springer. Evidence for a single loss of mineralized teeth in the common avian ancestor. *Science*, 346(6215):1254390, 2014. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1254390.

[212] D. Merkle and M. Middendorf. Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information. *Theory in Biosciences*, 123:277–299, 2005. ISSN 14317613. doi: 10.1016/j.thbio.2005.01.003.

[213] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):

1087–1092, 1953. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.1699114.

[214] R. Mihaescu, D. Levy, and L. Pachter. Why neighbor-joining works. *Algorithmica*, 54(1): 1–24, 2009. ISSN 0178-4617, 1432-0541. doi: 10.1007/s00453-007-9116-4.

[215] N. A. Moran and T. Jarvik. Lateral transfer of genes from fungi underlies carotenoid production in aphids. *Science*, 328(5978):624–627, 2010. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1187113.

[216] G. Moreno-Hagelsieb and K. Latimer. Choosing BLAST options for better detection of orthologs as reciprocal best hits. *Bioinformatics*, 24:319–324, 2008. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btm585.

[217] D. Morrison. Phylogenetic networks: A review of methods to display evolutionary history. *Annual Research & Review in Biology*, 4(10):1518–1543, 2014. ISSN 2347565X. doi: 10.9734/ARRB/2014/8230.

[218] E. A. Muchmore, S. Diaz, and A. Varki. A structural difference between the cell surfaces of humans and the great apes. *American Journal of Physical Anthropology*, 107(2):187–198, 1998. ISSN 0002-9483, 1096-8644. doi: 10.1002/(SICI)1096-8644(199810)107:2<187::AID-AJPA5>3.0.CO;2-S.

[219] A. Naor and U. Gophna. Cell fusion and hybrids in archaea: prospects for genome shuffling and accelerated strain development for biotechnology. *Bioengineered*, 4(3):126–129, 2013. ISSN 2165-5979, 2165-5987. doi: 10.4161/bioe.22649.

[220] A. Natanzon, R. Shamir, and R. Sharan. Complexity classification of some edge modification problems. *Discrete Applied Mathematics*, 113:109–128, 2001. ISSN 0166218X. doi: 10.1016/S0166-218X(00)00391-7.

[221] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48 (3):443–453, 1970. ISSN 00222836. doi: 10.1016/0022-2836(70)90057-4.

[222] N. L. Nehrt, W. T. Clark, P. Radivojac, and M. W. Hahn. Testing the ortholog conjecture with comparative functional genomic data from mammals. *PLoS Computational Biology*, 7:e1002073, 2011. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1002073.

[223] M. Nei and J. Zhang. Evolutionary distance: Estimation. In John Wiley & Sons, Ltd, editor, *Encyclopedia of Life Sciences*, page a0005108. John Wiley & Sons, Ltd, 2006. doi: 10.1038/npg.els.0005108.

[224] S. Nelson-Sathi, F. L. Sousa, M. Roettger, N. Lozada-Chávez, T. Thiergart, A. Janssen, D. Bryant, G. Landan, P. Schönheit, B. Siebers, J. O. McInerney, and W. F. Martin. Origins of major archaeal clades correspond to gene acquisitions from bacteria. *Nature*, 517:77–80, 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature13805.

[225] Y. Nevers, A. Kress, A. Defosset, R. Ripp, B. Linard, J. D. Thompson, O. Poch, and O. Lecompte. OrthoInspector 3.0: open portal for comparative genomics. *Nucleic Acids Research*, 47:D411–D418, 2019. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gky1068.

[226] B. T. L. Nichio, J. N. Marchaukoski, and R. T. Raittz. New tools in orthology analysis: A brief review of promising perspectives. *Frontiers in Genetics*, 8:165, 2017. ISSN 1664-8021. doi: 10.3389/fgene.2017.00165.

[227] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2000. ISBN 978-0-521-63235-5 978-0-521-63503-5.

[228] N. Nøjgaard, M. Geiß, D. Merkle, P. F. Stadler, N. Wieseke, and M. Hellmuth. Time-consistent reconciliation maps and forbidden time travel. *Algorithms for Molecular Biology*, 13:2, 2018. ISSN 1748-7188. doi: 10.1186/s13015-018-0121-8.

[229] P. S. Novichkov, M. V. Omelchenko, S. Gelfand Mikhail, A. A. Mironov, Y. I. Wolf, and E. V. Koonin. Genome-wide molecular clock and horizontal gene transfer in bacterial evolution. *Journal of Bacteriology*, 186:6575–6585, 2004. ISSN 0021-9193, 1098-5530. doi: 10.1128/JB.186.19.6575-6585.2004.

[230] C. Nüsslein-Volhard and E. Wieschaus. Mutations affecting segment number and polarity in drosophila. *Nature*, 287(5785):795–801, 1980. ISSN 0028-0836, 1476-4687. doi: 10.1038/287795a0.

[231] S. Ohno. *Evolution by Gene Duplication*. Springer Berlin Heidelberg, 1970. ISBN 978-3-642-86661-6 978-3-642-86659-3. doi: 10.1007/978-3-642-86659-3.

[232] M. V. Olson. When less is more: gene loss as an engine of evolutionary change. *The American Journal of Human Genetics*, 64(1):18–23, 1999. ISSN 00029297. doi: 10.1086/302219.

[233] M. V. Olson and A. Varki. Sequencing the chimpanzee genome: insights into human evolution and disease. *Nature Reviews Genetics*, 4(1):20–28, 2003. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg981.

[234] S. P. Otto and J. Whitton. Polyploid incidence and evolution. *Annual Review of Genetics*, 34:401–437, 2000. ISSN 0066-4197, 1545-2948. doi: 10.1146/annurev.genet.34.1.401.

[235] Y. Ovadia, D. Fielder, C. Conow, and R. Libeskind-Hadas. The cophylogeny reconstruction problem is NP-complete. *Journal of Computational Biology*, 18:59–65, 2011. ISSN 1066-5277, 1557-8666. doi: 10.1089/cmb.2009.0240.

[236] R. Overbeek, M. Fonstein, M. D'Souza, G. D. Pusch, and N. Maltsev. The use of gene clusters to infer functional coupling. *Proceedings of the National Academy of Sciences of the United States of America*, 96(6):2896–2901, 1999. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.96.6.2896.

[237] R. D. M. Page. Parallel phylogenies: reconstructing the history of host–parasite assemblages. *Cladistics*, 10:155–173, 1994. ISSN 0748-3007, 1096-0031. doi: 10.1111/j.1096-0031.1994.tb00170.x.

[238] R. D. M. Page and M. A. Charleston. Reconciled trees and incongruent gene and species trees. *DIMACS Ser Discrete Mathematics and Theor Comput Sci*, 37:57–70, 1997. doi: 10.1090/dimacs/037/04.

[239] D. Pan and L. Zhang. Tandemly arrayed genes in vertebrate genomes. *Comparative and Functional Genomics*, 2008:545269, 2008. ISSN 1531-6912, 1532-6268. doi: 10.1155/2008/545269.

[240] N. Panchy, M. Lehti-Shiu, and S.-H. Shiu. Evolution of gene duplication in plants. *Plant Physiology*, 171(4):2294–2316, 2016. ISSN 0032-0889, 1532-2548. doi: 10.1104/pp.16.00523.

[241] J. S. L. Patané, J. Martins, and J. C. Setubal. Phylogenomics. In J. C. Setubal, J. Stoye, and P. F. Stadler, editors, *Comparative Genomics*, volume 1704, pages 103–187. Springer New York, 2018. ISBN 978-1-4939-7461-0 978-1-4939-7463-4. doi: 10.1007/978-1-4939-7463-4_5. Series Title: Methods in Molecular Biology.

[242] W. R. Pearson. An introduction to sequence similarity ("homology") searching. *Current Protocols in Bioinformatics*, Chapter 3:Unit3.1, 2013. ISSN 1934-3396, 1934-340X. doi: 10.1002/0471250953.bi0301s42.

[243] M. N. Price, P. S. Dehal, and A. P. Arkin. FastTree 2 – approximately maximum-likelihood trees for large alignments. *PLoS ONE*, 5(3):e9490, 2010. ISSN 1932-6203. doi: 10.1371/journal.pone.0009490.

[244] K. D. Pruitt, T. Tatusova, and D. R. Maglott. NCBI reference sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research*, 33:D501–504, 2005. ISSN 1362-4962. doi: 10.1093/nar/gki025.

[245] B. Rannala and Z. Yang. Phylogenetic inference using whole genomes. *Annual Review of Genomics and Human Genetics*, 9:217–231, 2008. ISSN 1527-8204, 1545-293X. doi: 10.1146/annurev.genom.9.081307.164407.

[246] M. Ravenhall, N. Škunca, F. Lassalle, and C. Dessimoz. Inferring horizontal gene transfer. *PLOS Computational Biology*, 11:e1004095, 2015. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004095.

[247] R. Raz and A. Tal. Oracle separation of BQP and PH. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 13–23. ACM, 2019. ISBN 978-1-4503-6705-9. doi: 10.1145/3313276.3316315.

[248] J. A. Reuter, D. V. Spacek, and M. P. Snyder. High-throughput sequencing technologies. *Molecular Cell*, 58(4):586–597, 2015. ISSN 10972765. doi: 10.1016/j.molcel.2015.05.004.

[249] I. B. Rogozin, D. Managadze, S. A. Shabalina, and E. V. Koonin. Gene family level comparative analysis of gene expression in mammals validates the ortholog conjecture. *Genome Biology and Evolution*, 6(4):754–762, 2014. ISSN 1759-6653. doi: 10.1093/gbe/evu051.

[250] B. Rost. Twilight zone of protein sequence alignments. *Protein Engineering, Design and Selection*, 12(2):85–94, 1999. ISSN 1741-0134, 1741-0126. doi: 10.1093/protein/12.2.85.

[251] A. C. J. Roth, G. H. Gonnet, and C. Dessimoz. Algorithm of OMA for large-scale orthology inference. *BMC Bioinformatics*, 9:518, 2008. ISSN 1471-2105. doi: 10.1186/1471-2105-9-518.

[252] L. Y. Rusin, E. Lyubetskaya, K. Y. Gorbunov, and V. Lyubetsky. Reconciliation of gene and species trees. *BioMed Research International*, 2014:642089, 2014. ISSN 2314-6133, 2314-6141. doi: 10.1155/2014/642089.

[253] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987. ISSN 1537-1719. doi: 10.1093/oxfordjournals.molbev.a040454.

[254] D. Sánchez-Soto, V. Agüero-Chapin, Armijos-Jaramillo, Y. Perez-Castillo, E. Tejera, A. Antunes, and A. Sánchez-Rodríguez. ShadowCaster: Compositional methods under the shadow of phylogenetic models to detect horizontal gene transfers in prokaryotes. *Genes*, 11:756, 2020. ISSN 2073-4425. doi: 10.3390/genes11070756.

[255] F. Sanger, G. M. Air, B. G. Barrell, N. L. Brown, A. R. Coulson, J. C. Fiddes, C. A. Hutchison, P. M. Slocombe, and M. Smith. Nucleotide sequence of bacteriophage ϕx174 DNA. *Nature*, 265(5596):687–695, 1977. ISSN 0028-0836, 1476-4687. doi: 10.1038/265687a0.

[256] D. Sankoff. Minimal mutation trees of sequences. *SIAM Journal on Applied Mathematics*, 28(1):35–42, 1975.

[257] E. Sayyari and S. Mirarab. Testing for polytomies in phylogenetic species trees using quartet frequencies. *Genes*, 9:132, 2018. ISSN 2073-4425. doi: 10.3390/genes9030132.

[258] D. Schaller. Inference of best matches from evolutionary distance data. Master's thesis, University of Leipzig, Leipzig, Germany, 2019.

[259] D. Schaller, M. Geiß, E. Chávez, M. González Laffitte, A. López Sánchez, B. M. R. Stadler, D. I. Valdivia, M. Hellmuth, M. Hernández Rosales, and P. F. Stadler. Corrigendum to "Best match graphs". *Journal of Mathematical Biology*, 82:47, 2021. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-021-01601-6.

[260] D. Schaller, M. Geiß, M. Hellmuth, and P. F. Stadler. Best match graphs with binary trees. In C. Martín-Vide, M. A. Vega-Rodríguez, and T. Wheeler, editors, *Algorithms for Computational Biology, 8th AlCoB*, volume 12715 of *Lecture Notes in Computer Science*, pages 82–93, 2021. doi: 10.1007/978-3-030-74432-8_6.

[261] D. Schaller, M. Geiß, M. Hellmuth, and P. F. Stadler. Arc-completion of 2-colored best match graphs to binary-explainable best match graphs. *Algorithms*, 14(4):110, 2021. ISSN 1999-4893. doi: 10.3390/a14040110.

[262] D. Schaller, M. Geiß, M. Hellmuth, and P. F. Stadler. Least resolved trees for two-colored best match graphs. *Journal of Graph Algorithms and Applications*, 25(1):397–416, 2021. ISSN 1526-1719. doi: 10.7155/jgaa.00564.

[263] D. Schaller, M. Geiß, M. Hellmuth, and P. F. Stadler. Heuristic algorithms for best match graph editing. *Algorithms for Molecular Biology*, 16(1):19, 2021. ISSN 1748-7188. doi: 10.1186/s13015-021-00196-3.

[264] D. Schaller, M. Geiß, P. F. Stadler, and M. Hellmuth. Complete characterization of incorrect orthology assignments in best match graphs. *Journal of Mathematical Biology*, 82:20, 2021. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-021-01564-8.

[265] D. Schaller, M. Lafond, P. F. Stadler, N. Wieseke, and M. Hellmuth. Indirect identification of horizontal gene transfer. *Journal of Mathematical Biology*, 83(1):10, 2021. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-021-01631-0.

[266] D. Schaller, P. F. Stadler, and M. Hellmuth. Complexity of modification problems for best match graphs. *Theoretical Computer Science*, 865:63–84, 2021. ISSN 03043975. doi: 10.1016/j.tcs.2021.02.037.

[267] G. Schönknecht, W.-H. Chen, C. M. Ternes, G. G. Barbier, R. P. Shrestha, M. Stanke, A. Bräutigam, B. J. Baker, J. F. Banfield, R. M. Garavito, K. Carr, C. Wilkerson, S. A. Rensing, D. Gagneul, N. E. Dickenson, C. Oesterhelt, M. J. Lercher, and A. P. M. Weber. Gene transfer from bacteria and archaea facilitated evolution of an extremophilic eukaryote. *Science*, 339(6124):1207–1210, 2013. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1231707.

[268] F. Schreiber and E. L. L. Sonnhammer. Hieranoid: hierarchical orthology inference. *Journal of Molecular Biology*, 425(11):2072–2081, 2013. ISSN 00222836. doi: 10.1016/j.jmb.2013.02.018.

[269] C. R. Seemann and M. Hellmuth. The matroid structure of representative triple sets and triple closure computation. *European Journal of Combinatorics*, 70:384–407, 2018. ISSN 01956698. doi: 10.1016/j.ejc.2018.02.013.

[270] C. Semple. Reconstructing minimal rooted trees. *Discrete Applied Mathematics*, 127:489–503, 2003. ISSN 0166218X. doi: 10.1016/S0166-218X(02)00250-0.

[271] C. Semple and M. Steel. *Phylogenetics*, volume 24 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, UK, 2003.

[272] J. C. Setubal and P. F. Stadler. Gene phyologenies and orthologous groups. In J. C. Setubal, P. F. Stadler, and J. Stoye, editors, *Comparative Genomics*, volume 1704, pages 1–28. Springer, Heidelberg, 2018. doi: 10.1007/978-1-4939-7463-4_1.

[273] J. C. Setubal, N. F. Almeida, and A. R. Wattam. Comparative genomics for prokaryotes. *Methods Mol Biol*, 1704:55–78, 2018. doi: 10.1007/978-1-4939-7463-4_3.

[274] G. Sevillya, O. Adato, and S. Snir. Detecting horizontal gene transfer: a probabilistic approach. *BMC Genomics*, 21:106, 2020. ISSN 1471-2164. doi: 10.1186/s12864-019-6395-5.

[275] R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1-2):173–182, 2004. ISSN 0166218X. doi: 10.1016/j.dam.2004.01.007.

[276] B. J. Shapiro and M. F. Polz. Microbial speciation. *Cold Spring Harbor Perspectives in Biology*, 7(10):a018143, 2015. ISSN 1943-0264. doi: 10.1101/cshperspect.a018143.

[277] B. J. Shapiro, J.-B. Leducq, and J. Mallet. What is speciation? *PLOS Genetics*, 12(3): e1005860, 2016. ISSN 1553-7404. doi: 10.1371/journal.pgen.1005860.

[278] V. Sharma, N. Hecker, J. G. Roscito, L. Foerster, B. E. Langer, and M. Hiller. A genomics approach reveals insights into the importance of gene losses for mammalian adaptations. *Nature Communications*, 9(1):1215, 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-03667-1.

[279] A. J. Sharp, D. P. Locke, S. D. McGrath, Z. Cheng, J. A. Bailey, R. U. Vallente, L. M. Pertz, R. A. Clark, S. Schwartz, R. Segraves, V. V. Oseroff, D. G. Albertson, D. Pinkel, and E. E. Eichler. Segmental duplications and copy-number variation in the human genome. *The American Journal of Human Genetics*, 77(1):78–88, 2005. ISSN 00029297. doi: 10.1086/431652.

[280] Y. Shiloach and S. Even. An on-line edge-deletion problem. *Journal of the ACM*, 28:1–4, 1981. ISSN 0004-5411, 1557-735X. doi: 10.1145/322234.322235.

[281] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding, J. D. Thompson, and D. G. Higgins. Fast, scalable generation

of high-quality protein multiple sequence alignments using clustal omega. *Molecular Systems Biology*, 7:539, 2011. ISSN 1744-4292, 1744-4292. doi: 10.1038/msb.2011.75.

[282] M. Simonsen, T. Mailund, and C. N. S. Pedersen. Rapid neighbour-joining. In K. A. Crandall and J. Lagergren, editors, *Algorithms in Bioinformatics*, volume 5251, pages 113–122. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-87360-0 978-3-540-87361-7. doi: 10.1007/978-3-540-87361-7_10.

[283] J. Sjöstrand, A. Tofigh, V. Daubin, L. Arvestad, B. Sennblad, and J. Lagergren. A Bayesian method for analyzing lateral gene transfer. *Systematic Biology*, 63:409–420, 2014. ISSN 1063-5157, 1076-836X. doi: 10.1093/sysbio/syu007.

[284] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981. ISSN 00222836. doi: 10.1016/0022-2836(81)90087-5.

[285] R. R. Sokal and C. D. Michener. A statistical method for evaluating systematic relationships. *Univ. Kansas Science Bull.*, 38:1409–1438, 1958.

[286] C. Solís-Lemus, P. Bastide, and C. Ané. PhyloNetworks: A package for phylogenetic networks. *Molecular Biology and Evolution*, 34(12):3292–3298, 2017. ISSN 0737-4038, 1537-1719. doi: 10.1093/molbev/msx235.

[287] P. S. Soltis and D. E. Soltis. Ancient WGD events as drivers of key innovations in angiosperms. *Current Opinion in Plant Biology*, 30:159–165, 2016. ISSN 13695266. doi: 10.1016/j.pbi.2016.03.015.

[288] E. Sonnhammer, T. Gabaldón, A. Wilter Sousa da Silva, M. Martin, M. Robinson-Rechavi, B. Boeckmann, P. Thomas, C. Dessimoz, and Quest for Orthologs Consortium. Big data and other challenges in the quest for orthologs. *Bioinformatics*, 30:2993–2998, 2014. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btu492.

[289] E. L. L. Sonnhammer and G. Östlund. InParanoid 8: orthology analysis between 273 proteomes, mostly eukaryotic. *Nucleic Acids Research*, 43:D234–D239, 2015. ISSN 1362-4962, 0305-1048. doi: 10.1093/nar/gku1203.

[290] P. S. Soria, K. L. McGary, and A. Rokas. Functional divergence for every paralog. *Molecular Biology and Evolution*, 31:984–992, 2014. ISSN 1537-1719, 0737-4038. doi: 10.1093/molbev/msu050.

[291] S. M. Soucy, J. Huang, and J. P. Gogarten. Horizontal gene transfer: building the web of life. *Nature Reviews Genetics*, 16:472–482, 2015. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg3962.

[292] P. F. Stadler, S. J. Prohaska, C. V. Forst, and D. C. Krakauer. Defining genes: a computational framework. *Theory in Biosciences*, 128(3):165–170, 2009. ISSN 1431-7613, 1611-7530. doi: 10.1007/s12064-009-0067-y.

[293] P. F. Stadler, M. Geiß, D. Schaller, A. López Sánchez, M. Gonzalez Laffitte, D. I. Valdivia, M. Hellmuth, and M. Hernández Rosales. From pairs of most similar sequences to phylogenetic best matches. *Algorithms for Molecular Biology*, 15:5, 2020. ISSN 1748-7188. doi: 10.1186/s13015-020-00165-2.

[294] T. Stadler. Simulating trees with a fixed number of extant species. *Systematic Biology*, 60(5):676–684, 2011. ISSN 1076-836X, 1063-5157. doi: 10.1093/sysbio/syr029.

[295] A. Stamatakis. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313, 2014. ISSN 1460-2059, 1367-4803. doi: 10.1093/bioinformatics/btu033.

[296] M. Stamboulian, R. F. Guerrero, M. W. Hahn, and P. Radivojac. The ortholog conjecture revisited: The value of orthologs and paralogs in function prediction. *Bioinformatics*, 36:i219–i226, 2020. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btaa468.

[297] T. B. Stanton. Prophage-like gene transfer agents-novel mechanisms of gene exchange for methanococcus, desulfovibrio, brachyspira, and rhodobacter species. *Anaerobe*, 13(2):43–49, 2007. ISSN 10759964. doi: 10.1016/j.anaerobe.2007.03.004.

[298] H. H. Stedman, B. W. Kozyak, A. Nelson, D. M. Thesier, L. T. Su, D. W. Low, C. R. Bridges, J. B. Shrager, N. Minugh-Purvis, and M. A. Mitchell. Myosin gene mutation correlates with anatomical changes in the human lineage. *Nature*, 428(6981):415–418, 2004. ISSN 0028-0836, 1476-4679. doi: 10.1038/nature02358.

[299] M. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*, 9(1):91–116, 1992. ISSN 0176-4268, 1432-1343. doi: 10.1007/BF02618470.

[300] M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal of the ACM*, 44(4):585–591, 1997. ISSN 0004-5411, 1557-735X. doi: 10.1145/263867.263872.

[301] C. E. V. Storm and E. L. L. Sonnhammer. Automated ortholog inference from phylogenetic trees and calculation of orthology reliability. *Bioinformatics*, 18(1):92–99, 2002. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/18.1.92.

[302] M. A. Suchard, P. Lemey, G. Baele, D. L. Ayres, A. J. Drummond, and A. Rambaut. Bayesian phylogenetic and phylodynamic data integration using BEAST 1.10. *Virus Evolution*, 4(1), 2018. ISSN 2057-1577. doi: 10.1093/ve/vey016.

[303] E. Susko and A. J. Roger. Long branch attraction biases in phylogenetics. *Systematic Biology*, 2021. ISSN 1063-5157, 1076-836X. doi: 10.1093/sysbio/syab001.

[304] K. M. Swenson, A. Doroftei, and N. El-Mabrouk. Gene tree correction for reconciliation and species tree inference. *Algorithms for Molecular Biology*, 7:31, 2012. ISSN 1748-7188. doi: 10.1186/1748-7188-7-31.

[305] D. L. Swofford, G. J. Olsen, P. J. Waddell, and D. M. Hillis. Phylogenetic inference. In D. M. Hillis, C. Moritz, and B. K. Mable, editors, *Molecular Systematics*, pages 407–514. Sinauer Associates, Sunderland (MA), 2nd edition, 1996.

[306] G. J. Szöllősi, B. Boussau, S. S. Abby, E. Tannier, and V. Daubin. Phylogenetic modeling of lateral gene transfer reconstructs the pattern and relative timing of speciations. *Proceedings of the National Academy of Sciences*, 109(43):17513–17518, 2012. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1202997109.

[307] K. Takahashi, Y. Terai, M. Nishida, and N. Okada. Phylogenetic relationships and ancient incomplete lineage sorting among cichlid fishes in Lake Tanganyika as revealed by analysis of the insertion of retroposons. *Molecular Biology and Evolution*, 18:2057–2066, 2001. ISSN 0737-4038, 1537-1719. doi: 10.1093/oxfordjournals.molbev.a003747.

[308] E. L. Tatum and J. Lederberg. Gene recombination in the bacterium escherichia coli. *Journal of Bacteriology*, 53(6):673–684, 1947. ISSN 0021-9193, 1098-5530. doi: 10.1128/JB.53.6.673-684.1947.

[309] R. L. Tatusov, E. V. Koonin, and D. J. Lipman. A genomic perspective on protein families. *Science*, 278:631–637, 1997. ISSN 00368075, 10959203. doi: 10.1126/science.278.5338.631.

[310] 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 Research*, 28: 33–36, 2000. ISSN 13624962. doi: 10.1093/nar/28.1.33.

[311] S. Tavaré. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on Mathematics in the Life Sciences*, 17:57–86, 1986.

[312] S. J. Tazehkand, S. N. Hashemi, and H. Poormohammadi. New heuristics for rooted triplet consistency. *Algorithms*, 6:396–406, 2013. ISSN 1999-4893. doi: 10.3390/a6030396.

[313] A. I. Teufel, L. Liu, and D. A. Liberles. Models for gene duplication when dosage balance works as a transition state to subsequent neo-or sub-functionalization. *BMC Evolutionary Biology*, 16:45, 2016. ISSN 1471-2148. doi: 10.1186/s12862-016-0616-1.

[314] The New York Times. Coronavirus variants and mutations, 2021. URL `https://www.nytimes.com/interactive/2021/health/coronavirus-variant-tracker.html`.

[315] C. M. Thomas and K. M. Nielsen. Mechanisms of, and barriers to, horizontal gene transfer between bacteria. *Nature Reviews Microbiology*, 3:711–721, 2005. ISSN 1740-1526, 1740-1534. doi: 10.1038/nrmicro1234.

[316] A. Tofigh, M. Hallett, and J. Lagergren. Simultaneous identification of duplications and lateral gene transfers. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(2):517–535, 2011. ISSN 1545-5963. doi: 10.1109/TCBB.2010.14.

[317] V. A. Traag. Faster unfolding of communities: Speeding up the louvain algorithm. *Physical Review E*, 92(3):032801, 2015. ISSN 1539-3755, 1550-2376. doi: 10.1103/PhysRevE.92.032801.

[318] C.-M. Train, N. M. Glover, G. H. Gonnet, A. M. Altenhoff, and C. Dessimoz. Orthologous matrix (OMA) algorithm 2.0: more robust to asymmetric evolutionary rates and more scalable hierarchical orthologous group inference. *Bioinformatics*, 33:i75–i82, 2017. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btx229.

[319] D. Tsur. Faster algorithms for cograph edge modification problems. *Information Processing Letters*, 158:105946, 2020. ISSN 00200190. doi: 10.1016/j.ipl.2020.105946.

[320] I. Ullah, J. Sjöstrand, P. Andersson, B. Sennblad, and J. Lagergren. Integrating sequence evolution into probabilistic orthology analysis. *Systematic Biology*, 64(6):969–982, 2015. ISSN 1063-5157, 1076-836X. doi: 10.1093/sysbio/syv044.

[321] L. van Dorp, M. Acman, D. Richard, L. P. Shaw, C. E. Ford, L. Ormond, C. J. Owen, J. Pang, C. C. S. Tan, F. A. T. Boshier, A. T. Ortiz, and F. Balloux. Emergence of genomic diversity and recurrent mutations in SARS-CoV-2. *Infection, Genetics and Evolution: Journal of Molecular Epidemiology and Evolutionary Genetics in Infectious Diseases*, 83:104351, 2020. ISSN 1567-7257. doi: 10.1016/j.meegid.2020.104351.

[322] J. van Wyhe (ed.). The complete work of Charles Darwin online, 2002. URL `http://darwin-online.org.uk/`.

[323] B. Vernot, M. Stolzer, A. Goldman, and D. Durand. Reconciliation with non-binary species trees. *Journal of Computational Biology*, 15:981–1006, 2008. ISSN 1066-5277, 1557-8666. doi: 10.1089/cmb.2008.0092.

[324] 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 Research*, 19(2):327–335, 2008. ISSN 1088-9051. doi: 10.1101/gr.073585.107.

[325] D. P. Wall and T. DeLuca. Ortholog detection using the reciprocal smallest distance algorithm. In N. H. Bergman, editor, *Comparative Genomics*, volume 396 of *Methods in Molecular Biology*, pages 95–110. Humana Press, 2007. ISBN 978-1-934115-37-4 978-1-59745-515-2. doi: 10.1007/978-1-59745-515-2_7.

[326] I. Wapinski, A. Pfeffer, N. Friedman, and A. Regev. Automatic genome-wide reconstruction of phylogenetic gene trees. *Bioinformatics*, 23(13):i549–558, 2007. ISSN 1460-2059, 1367-4803. doi: 10.1093/bioinformatics/btm193.

[327] F. Weiling. Historical study: Johann gregor mendel 1822-1884. *American Journal of Medical Genetics*, 40(1):1–25; discussion 26, 1991. ISSN 0148-7299. doi: 10.1002/ajmg.1320400103.

[328] D. Wen, Y. Yu, J. Zhu, and L. Nakhleh. Inferring phylogenetic networks using PhyloNet. *Systematic Biology*, 67(4):735–740, 2018. ISSN 1063-5157, 1076-836X. doi: 10.1093/sysbio/syy015.

[329] S. Whelan. Inferring trees. In J. M. Keith, editor, *Bioinformatics*, volume 452 of *Methods in Molecular Biology*, pages 287–309. Humana Press, 2008. ISBN 978-1-58829-707-5 978-1-60327-159-2. doi: 10.1007/978-1-60327-159-2_14.

[330] N. Wieseke, M. Bernt, and M. Middendorf. Unifying parsimonious tree reconciliation. In A. Darling and J. Stoye, editors, *Algorithms in Bioinformatics. WABI 2013*, volume 8126 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, 2013. Springer. doi: 10.1007/978-3-642-40453-5_16.

[331] D. Williams, J. P. Gogarten, and R. T. Papke. Quantifying homologous replacement of loci between haloarchaeal species. *Genome Biology and Evolution*, 4:1223–1244, 2012. ISSN 1759-6653. doi: 10.1093/gbe/evs098.

[332] P. Worning, L. J. Jensen, K. E. Nelson, S. Brunak, and D. W. Ussery. Structural analysis of DNA sequence: evidence for lateral gene transfer in thermotoga maritima. *Nucleic Acids Research*, 28(3):706–709, 2000. ISSN 13624962. doi: 10.1093/nar/28.3.706.

[333] B. Y. Wu. Constructing the maximum consensus tree from rooted triples. *Journal of Combinatorial Optimization*, 8:29–39, 2004. ISSN 1382-6905. doi: 10.1023/B:JOCO.0000021936.04215.68.

[334] Z. Yang. Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Molecular Biology and Evolution*, 10(6):1396–1401, 1993. ISSN 1537-1719. doi: 10.1093/oxfordjournals.molbev.a040082.

[335] Z. Yang. *Computational molecular evolution*. Oxford series in ecology and evolution. Oxford University Press, 2006. ISBN 978-0-19-856699-1 978-0-19-856702-8.

[336] Z. Yang. *Molecular evolution: a statistical approach*. Oxford University Press, first edition edition, 2014. ISBN 978-0-19-960261-2 978-0-19-960260-5.

[337] M. Yannakakis. Computing the Minimum Fill-In is NP-complete. *SIAM Journal on Algebraic Discrete Methods*, 2:77–79, 1981. ISSN 0196-5212, 2168-345X. doi: 10.1137/0602010.

[338] A. D. Yates, P. Achuthan, W. Akanni, J. Allen, J. Allen, J. Alvarez-Jarreta, M. R. Amode, I. M. Armean, A. G. Azov, R. Bennett, J. Bhai, K. Billis, S. Boddu, J. C. Marugán, C. Cummins, C. Davidson, K. Dodiya, R. Fatima, A. Gall, C. G. Giron, L. Gil, T. Grego, L. Haggerty, E. Haskell, T. Hourlier, O. G. Izuogu, S. H. Janacek, T. Juettemann, M. Kay, I. Lavidas, T. Le, D. Lemos, J. G. Martinez, T. Maurel, M. McDowall, A. McMahon, S. Mohanan, B. Moore, M. Nuhn, D. N. Oheh, A. Parker, A. Parton, M. Patricio, M. P. Sakthivel, A. I. Abdul Salam, B. M. Schmitt, H. Schuilenburg, D. Sheppard, M. Sycheva, M. Szuba, K. Taylor, A. Thormann, G. Threadgold, A. Vullo, B. Walts, A. Winterbottom, A. Zadissa, M. Chakiachvili, B. Flint, A. Frankish, S. E. Hunt, G. IIsley, M. Kostadima, N. Langridge, J. E. Loveland, F. J. Martin, J. Morales, J. M. Mudge, M. Muffato, E. Perry, M. Ruffier, S. J. Trevanion, F. Cunningham, K. L. Howe, D. R. Zerbino, and P. Flicek. Ensembl 2020. *Nucleic Acids Research*, 48:D682–D688, 2020. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkz966.

[339] G. U. Yule. A mathematical theory of evolution, based on the conclusions of dr. j. c. willis, f. r. s. *Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of a Biological Character*, 213(402):21–87, 1924. ISSN 0264-3960, 2053-9266. doi: 10.1098/rstb.1925.0002.

[340] R. Zallot, K. J. Harrison, B. Kolaczkowski, and V. de Crécy-Lagard. Functional annotations of paralogs: A blessing and a curse. *Life*, 6:39, 2016. ISSN 2075-1729. doi: 10.3390/life6030039.

[341] J. Zhang. Evolution by gene duplication: an update. *Trends in Ecology & Evolution*, 18(6):292–298, 2003. ISSN 01695347. doi: 10.1016/S0169-5347(03)00033-8.

[342] L. Zhang. On a mirkin-muchnik-smith conjecture for comparing molecular phylogenies. *Journal of Computational Biology*, 4(2):177–187, 1997. ISSN 1066-5277, 1557-8666. doi: 10.1089/cmb.1997.4.177.

[343] E. Zuckerkandl and L. Pauling. *Molecular Disease, Evolution, and Genic Heterogeneity*. Academic Press, 1962.

[344] I. E. Zverovich. Near-complete multipartite graphs and forbidden induced subgraphs. *Discrete Mathematics*, 207:257–262, 1999. ISSN 0012365X. doi: 10.1016/S0012-365X(99)00050-3.

CURRICULUM VITAE

PERSONAL INFORMATION

Name	David Schaller
Date of birth	June 1, 1994
Place of birth	Glauchau, Germany
Email	sdavid@bioinf.uni-leipzig.de
GitHub	david-schaller

EDUCATION

11/2019–06/2021	**PhD student** Bioinformatics group, Leipzig University Max Planck Institute for Mathematics in the Sciences, Leipzig
10/2017–10/2019	**Master studies in Bioinformatics** Leipzig University Finished with **Master of Science** (final grade: 1.0) Thesis: Inference of Best Matches from Evolutionary Distance Data
10/2014–09/2017	**Bachelor studies in Biology** Leipzig University Finished with **Bachelor of Science** (final grade: 1.0) Thesis: Analyse von mechanisch gestreckten Keratinozyten
08/2005–07/2013	**Secondary school** Georgius-Agricola-Gymnasium, Glauchau Finished with **Abitur** (final grade: 1.0)
08/2008–07/2012	**Professional school** (parallel professional training) BSZ "Dr. Friedrich Dittes", Glauchau Qualification **"Assistant for Computer Science"**

TEACHING

Practical course

Graphs and biological networks (Leipzig University, winter semester 2020/21)

Supervised Bachelor thesis

Mohammed Habib: Exact & heuristic algorithms for correcting disturbed Later-Divergence-Time Graphs (Stockholm University, August 2021, co-supervision with Prof. Marc Hellmuth)

LANGUAGES

German (native), English (fluent), French (advanced)

COMPUTER SKILLS

Programming	Python, C++, Java, R, Perl
Misc	LaTeX, Microsoft Office, Inkscape, Photoshop

LIST OF PUBLICATIONS

D. Schaller, M. Hellmuth, P.F. Stadler (2021) "A Simple Linear-Time Algorithm for the Common Refinement of Rooted Phylogenetic Trees on a Common Leaf Set." Submitted to: *Algorithms for Molecular Biology*, arXiv:2107.00072

A. Korchmaros, D. Schaller, M. Hellmuth, P.F. Stadler (2021) "Quasi-Best Match Graphs." Submitted to: *Discrete Applied Mathematics*, arXiv:2109.10235

M. Hellmuth, M. Michel, N. Nøgaard, D. Schaller, P.F. Stadler (2021) "Combining Orthology and Xenology Data in a Common Phylogenetic Tree." Accepted: *Brazilian Symposium on Bioinformatics, BSB 2021*, arXiv:2107.01893

M. Hellmuth, D. Schaller, P.F. Stadler (2021). "Compatibility of Partitions, Hierarchies, and Split Systems." Submitted to: *Discrete Applied Mathematics*, arXiv:2104.14146

D. Schaller, M. Geiß, M. Hellmuth, P.F. Stadler (2021). "Heuristic Algorithms for Best Match Graph Editing." In: *Algorithms for Molecular Biology*, 16(1):19, doi: 10.1186/s13015-021-00196-3

D. Schaller, M. Geiß, M. Hellmuth, P.F. Stadler (2021). "Least resolved trees for two-colored best match graphs." In: *Journal of Graph Algorithms and Applications*, 25(1):397-416, doi: 10.7155/jgaa.00564

D. Schaller, M. Lafond, P.F. Stadler, N. Wieseke, M. Hellmuth (2021). "Indirect Identification of Horizontal Gene Transfer." In: *Journal of Mathematical Biology*, 83(1):10, doi: 10.1007/s00285-021-01631-0

D. Schaller, M. Geiß, M. Hellmuth, P.F. Stadler (2021). "Best Match Graphs with Binary Trees." In: C. Martín-Vide, M.A. Vega-Rodríguez, T. Wheeler (Eds.), *Algorithms for Computational Biology, 8th AlCoB*, volume 12715 of *Lecture Notes in Computer Science*, pages 82-93, doi: 10.1007/978-3-030-74432-8_6

D. Schaller, M. Geiß, M. Hellmuth, P.F. Stadler (2021). "Arc-Completion of 2-Colored Best Match Graphs to Binary-Explainable Best Match Graphs." In: *Algorithms*, 14(4):110, doi: 10.3390/a14040110

D. Schaller, M. Geiß, E. Chávez, M.G. Lafitte, A. López Sánchez, B.M. Stadler, D.I. Valdivia, M. Hellmuth, M. Hernández-Rosales, P.F. Stadler (2021). "Corrigendum to "Best match graphs"." In: *Journal of Mathematical Biology*, 82(6):47, doi: 10.1007/s00285-021-01601-6

D. Schaller, P.F. Stadler, M. Hellmuth (2021). "Complexity of modification problems for best match graphs." In: *Theoretical Computer Science*, 865:63-84, doi: 10.1016/j.tcs.2021.02.037

D. Schaller, M. Geiß, P.F. Stadler, M. Hellmuth (2021). "Complete Characterization of Incorrect Orthology Assignments in Best Match Graphs." In: *Journal of Mathematical Biology*, 82(3):20, doi: 10.1007/s00285-021-01564-8

P.F. Stadler, M. Geiß, D. Schaller, A. López Sánchez, M. González Laffitte, D.I. Valdivia, M. Hellmuth, M. Hernández-Rosales (2020). "From pairs of most similar sequences to phylogenetic best matches." In: *Algorithms for Molecular Biology*, 15:5, doi: 10.1007/s12064-009-0067-y

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 Materialen oder erbrachten Dienstleistungen als solche gekennzeichnet.

Leipzig, 26. Oktober 2021

David Schaller