From Best Match Graphs to Gene Trees

A new perspective on graph-based orthology inference

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

D I S S E R T A T I O N

zur Erlangung des akademischen Grades

Doctor rerum naturalium
(Dr. rer. nat.)

im Fachgebiet

Informatik

vorgelegt von

Manuela Geiß, BSc. BSc. MSc.

geboren am 18.03.1990 in Bad Soden am Taunus, Deutschland

Die Annahme der Dissertation wurde empfohlen von:

1. Prof. Dr. Peter F. Stadler, Universität Leipzig
2. Prof. Dr. Vincent Moulton, University of East Anglia

Die Verleihung des akademischen Grades erfolgt mit Bestehen
der Verteidigung am 29.10.2019 mit dem Gesamtprädikat *summa cum laude*.

ACKNOWLEDGEMENTS

First of all, I want to express my warmest gratitude to Peter Stadler for the possibility to contribute to this awesome research project, his interest for my work, his trust and faith in me, and many many interesting and productive discussions.
Furthermore, many thanks to (my somehow unofficial second supervisor ;-)) Marc Hellmuth for the close collaboration during my PhD, all the support and help as well as many endless discussions about all the details of my proofs - I learned a lot from you!
I would also like to thank Sven Findeiß who paved the way for me to go to Leipzig, went through the ups and downs of teaching with me, and made cakes magically appear on my desk in the morning.
Of course, many thanks to all my colleagues at the Bioinf for creating such an exceptionally pleasant working environment at the institute.
I also like to thank the MexMafia around Maribel Hernandez-Rosales for good collaboration and a wonderful time in Mexico.
Last but not least, I want to thank my family and in particular my parents for their never ending support in all situations of life, and Stefan for always being there for me.

ABSTRACT

Orthology detection is an important task within the context of genome annotation, gene nomenclature, and the understanding of gene evolution. With the rapidly accelerating pace at which new genomes become available, highly efficient methods are urgently required. As demonstrated in a large body of literature, reciprocal best match (RBH) methods are reasonably accurate and scale to large data sets. Nevertheless, they are far from perfect and prone to both, false positive and false negative, orthology calls.

This work gives a complete characterization of best match as well as reciprocal best match graphs (BMGs and RBMGs) that arise at the first step of RBH methods. While BMGs as well as RBMGs with at most three species can be recognized in polynomial time, RBMGs with more than three species have a surprisingly complicated structure and it remains an open problem whether there exist polynomial time algorithms for the recognition of these RBMGs. In contrast to RBMGs, for which many (often mutually inconsistent) least resolved trees may exist, there is a unique least resolved tree for BMGs. This tree is a homeomorphic image of the true, but typically unknown, gene tree.

Furthermore, in the absence of horizontal gene transfer (HGT), the reciprocal best match graph contains the orthology relation suggesting that RBMGs can only contain false positive but no false negative orthology assignments. Simulation scenarios reveal that so-called good quartets, a certain graph pattern on four vertices in BMGs, can be used to successfully identify almost all false positive edges in RBMGs. Together with the existence of a unique least resolved tree, this suggests that BMGs contain a lot of valuable information for orthology inference that would be lost by exclusively considering RBMGs. These insights motivate to include additional BMG and RBMG editing steps in orthology detection pipelines based on the presented theoretical insights.

Moreover, a workflow is introduced to infer best matches from sequence data by retrieving quartet structures from local information instead of reconstructing the whole gene tree. A crucial prerequisite for this pipeline is the choice of suitable outgroups.

However, the empirical simulations also reveal that HGT events cause strong deviations of the orthology relation from the RBMG as well as good quartets that are no longer associated with false positive orthologs, suggesting the need for further investigation of the xenology relation.

The directed Fitch's xenology relation is characterized in terms of forbidden 3-vertex subgraphs and moreover, a polynomial time algorithm for the recognition and the reconstruction of a unique least resolved tree is presented. The undirected Fitch relation, in contrast, is shown to be a complete multipartite graph, which does not provide any interesting phylogenetic information.

In summary, the results of this work can be used to develop new methods for inferring orthology, paralogy, and HGT. They promise major improvements in the accuracy and the computational performance of RBH-based approaches.

CONTENTS

PUBLICATIONS

Some results and figures in this thesis have previously been published in the following publications:

1. M. Geiß, J. Anders, P. F. Stadler, N. Wieseke, and M. Hellmuth (2018). "Reconstructing Gene Trees from Fitch's Xenology Relation." In: *Journal of Mathematical Biology*, 77, (5), pp. 1459 - 1491

2. M. Hellmuth, Y. Long, M. Geiß, and P. F. Stadler (2018). "A Short Note on Undirected Fitch Graphs." In: *The Art of Discrete and Applied Mathematics*, 1, (1), pp. #P1.08

3. M. Geiß, E. Chávez, M. González Laffitte, A. López Sánchez, B. M. R. Stadler, D. I. Valdivia, M. Hellmuth, M. Hernandez Rosales, and P. F. Stadler (2019). "Best Match Graphs." In: *Journal of Mathematical Biology*, 78, (7), pp. 2015–2057

4. M. Geiß, P. F. Stadler, and M. Hellmuth (2019). "Reciprocal Best Match Graphs." Submitted to: *Journal of Mathematical Biology*, arxiv q-bio 1903.07920

5. M. Geiß, M. González Laffitte, A. López Sánchez, D. I. Valdivia, M. Hellmuth, M. Hernandez Rosales, and P. F. Stadler (2019). "Best Match Graphs and Reconciliation of Gene Trees with Species Trees." Submitted to: *Journal of Mathematical Biology*, arxiv q-bio 1904.12021

6. P. F. Stadler, M. Geiß, D. Schaller, A. López Sánchez, M. González Laffitte, D. I. Valdivia, M. Hellmuth, and M. Hernandez Rosales (2019). "From Best Hits to Best Matches." Submitted to: *23th Conference on Algorithmic Computational Biology (RECOMB 2019)*

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

1

PREFACE

"Nothing in biology makes sense except in the light of evolution."

This statement by Theodosius Dobzhansky from the early 1970s [51] is one of the most famous citations in the field of evolutionary biology and embodies the continuously growing relevance of this field. Losos et al. [152] even claim that "the next 20 years hold the promise of a golden age for evolutionary biology." Indeed, as explicitly stated, evolutionary concepts and analyses play a central role in the four broad challenges for biology identified by the 2009 report *A New Biology for the 21st Century* commissioned by the National Research Council of the National Academies [39]. These four challenges are the development of sustainably growing plants for efficient food production, understanding and sustaining ecosystems and biodiversity, the expansion of renewable energies, and understanding individual's health.

Phylogenetics is a subfield of evolutionary biology that studies the evolutionary relationships and history of biological entities, mainly individuals or groups of organisms such as populations or species, by comparison of specific heritable traits. These relationships are typically represented by phylogenetic trees or networks. Besides some fossil records, however, there is a lack of knowledge about the past and in particular about extinct species, hence phylogenetics can infer hypotheses about evolutionary history from extant species only [168].

In early phylogenetics, evolutionary relationships have been mainly deduced from morphological and physiological traits. This approach succeeded in inferring the major facets of evolutionary history, however, due to the complexity of such traits, often failed in disentangling the details, e.g. on a level of evolutionary relationship between closely related species [168]. At that time phylogenetics has been almost exclusively used in *systematics* and *taxonomy* [234]. Nowadays phylogenetics is present in almost every branch of biology with applications ranging from disentangling evolutionary history of gene families, species, or populations [186, 195], analyzing and decelerating the evolution of antibiotic resistance [11, 180] to language evolution [67, 192]. Moreover, with the rapidly increasing amount of available genomic data, whose collection becomes continuously cheaper and easier due to technological advances, *molecular phylogenetics* has become indispensable for comparative genomics, e.g. for the identification of genes, non-coding RNAs or other regulatory elements in newly sequences genomes [127, 181], the prediction of structure-function relationship [223], or the reconstruction of ancestral genomes [166].

While *species* in molecular phylogenetics are characterized by their *genome*, i.e., their complete set of DNA or RNA, a *gene* is considered as a part of the genome at a certain position. This might for instance be a protein coding region, a region of non-coding RNA, or a regulatory region such as a promotor or an enhancer. Genes from distinct species that share the same ancestry, i.e., have

emanated from a common origin, form a *gene family* and are called *homologs* [64].

The underlying idea of evolutionary theory is the existence of one origin of life, i.e., all organisms have emerged from one common ancestor [43]. Although the genetic variation of today's living organisms has been caused by many mechanisms such as different types of mutations, genome rearrangements, and gene exchange [146], phylogenetic approaches are mainly concerned with four evolutionary events:
(i) *Speciation*: divergence of a species into two or more descendant species,
(ii) *Gene duplication*: a genomic region is copied within the genome,
(iii) *Horizontal gene transfer*: exchange of genetic material among co-existing species,
(iv) *Gene loss*: extinction of a gene.
Two homologous genes are called *orthologs* if their last common ancestor was a speciation event. Likewise, if two genes have evolved from a gene duplication or if horizontal gene transfer occurred, they are called *paralogs* and *xenologs*, respectively [66]. A formal definition of this terminology as well as a mathematical formulation of these concepts in terms of binary relations will be given in Chapters 2, 6, and 8, respectively.

The evolutionary history of a set of species is considered as a process starting from the common origin of these species, where genetic material is transferred from one generation to the next and mutations accumulate over time in the genome. These mutations increase the genetic variation between different populations of a species and eventually lead to species divergence [146]. This results in a tree-like evolution of species. The evolution of genes clearly follows the evolution of the corresponding species, i.e., the gene tree can be embedded into the species tree. More precisely, such an embedding corresponds to a mapping between gene and species tree, a so-called *reconciliation map* or simply *reconciliation* (see Chapter 2 for a formal definition).

Providing the theoretical framework and developing algorithms for disentangling and analyzing the evolutionary history of genes and species is the task of mathematical phylogenetics. The main focus here lies on the reconstruction of gene and species trees, and a reconciliation between those as well as on the inference of orthology, paralogy, and xenology events.

The first step of phylogenomic studies that are concerned with the reconciliation of gene and species tree, is typically the reconstruction of a species tree. Since gene and species trees need not necessarily to be congruent due to gene duplication coupled with loss or horizontal gene transfer, the reconstruction of species trees is based on orthologous genes, i.e., gene families that are suspected to contain paralogs or xenologs are excluded from the analysis. Trees for the single gene trees are then build in the next step and reconciled with the species tree in order to infer orthology, paralogy, and xenology [153, 194]. As a by-product, one obtains the assignment of evolutionary events to the inner nodes of the gene tree. Note in this context that gene loss cannot be directly inferred from extant genes. There exist approaches to infer gene loss indirectly, however, most of them being parsimonious [77, 153] (see also Section 2.3). Although having been widely neglected as an evolutionary force in the past, the

increasing amount of genomic data suggests that gene loss plays an important role in evolution [5, 121].

This work is a contribution to a second class of more recently developed methods that first estimate orthology, paralogy, and xenology relations directly from genomic data and then, in a second step, infer a gene tree, a species trees as well as a reconciliation map between them from those estimated relations. In this context, the evolutionary relationships are translated into the mathematical concept of a graph. Based on the fact that two orthologous genes form a pair of reciprocally most closely related genes, estimates for orthology are typically obtained from pairwise sequence comparison by so-called *best matches* and *reciprocal best matches*. Extensive benchmarking has shown that methods based on (reciprocal) best matches perform at least as well as the classical tree reconciliation approaches [10, 169]. However, such best match methods are far from perfect. Besides errors in the initial (reciprocal) best matches, false orthology assignments can, for instance, be caused by large differences in the evolutionary rate between paralogs [68]. This raises the question:
What is the mathematical structure of best match graphs and reciprocal best match graphs and how much information on the gene tree is contained in these relations?
Before using these insights about the structure of (reciprocal) best match graphs for correcting the initial erroneous estimates, we need to answer the next questions:
How are (reciprocal) best match graphs related to the orthology relation and to what extend can the orthology relation be inferred from those graphs? How can best match estimates be retrieved from sequence data?
As we shall see, the correct inference of orthology and paralogy as well as the editing of the initial estimates highly depends on whether horizontal gene transfer is present or not, it is crucial to understand horizontal gene transfer and the so-called *Fitch's Xenology Relation* (see Chapter 3) in more detail, which begs the question:
What is the mathematical structure of the xenology relation? How much information on the gene tree and the position of horizontal transfer events within the gene tree is contained in the xenology relation? How can this information be efficiently extracted from the xenology relation?

These questions are answered in this work by developing mathematical characterizations for these binary relations and giving algorithms for the recognition of these relations as well as for the reconstruction of a corresponding tree that represents a given relation. Moreover, simulation results are presented to answer the questions about the relationship between reciprocal best match graphs and the orthology relation.

STRUCTURE OF THIS THESIS

Chapter 2 formally introduces the concepts of homology, orthology, paralogy, and xenology, and gives an overview of well-known results about the orthology relation. In particular, it is discussed how evolutionary events can be inferred from sequence data. State-of-the-art methods for both approaches, i.e., either

the direct tree-based approach or the indirect graph-based approach are presented. Basic definitions and some well-known results used in this work are given in Chapter 3.
Mathematical characterizations of the best match relation and the reciprocal best match relation are developed in Chapters 4 and 5, respectively. Furthermore, recognition algorithms for these relations as well as reconstruction algorithms for the corresponding tree representations are given. Chapter 5, in addition, discusses in detail one class of reciprocal best match graphs that has the structure of an orthology relation. Chapter 6 then connects (reciprocal) best match graphs to the reconciliation of gene and species tree. Apart from theoretical findings, this chapter also presents simulation results which show to what extend the theoretical insights can be used to infer the correct orthology relation. The question how and under what conditions best matches can be correctly inferred from data is tackled in Chapter 7. Finally, the horizontal gene transfer relation is treated in Chapter 8, which gives a mathematical characterization as well as recognition and tree reconstruction algorithms.

2 INTRODUCTION

Phylogenetics makes hypotheses about the evolutionary history of biological entities, such as genes and species, by reconstructing a phylogenetic tree (or network) that represents the evolutionary history of a subset of genes from a gene family or a set of species starting from a common ancestor. While the evolution of gene families is usually assumed to be tree-like, species may also evolve along a phylogenetic network, which reflects scenarios such as hybridization between species [156]. In this thesis, we consider the simplified model of a tree-like species evolution. We refer to [114] for a detailed survey on phylogenetic networks and their inference from sequence data.

The evolution of a set of genes from a common gene family is represented by a so-called *gene tree*. The leaves of such a gene tree correspond to the extant genes, i.e., genes that reside within extant species, while inner vertices of the tree refer to genes from ancestral species. The root of the gene tree represents the common origin of all genes under consideration. Note in this context that many tree inference methods reconstruct unrooted trees, reflecting the fact that the common origin of a set of homologous genes cannot always be identified with certainty [146]. The focus of this work, however, lies on rooted trees.

The evolutionary history of a particular extant gene x starting at the common origin of the gene family under consideration is given by a sequence of divergence events, more precisely by the unique path from the root of the gene tree to the leaf x, which consists of a sequence of speciation, duplication, and HGT events. The progenitor genes along this path are called *ancestors* of x, whereas x is a descendant of each its ancestors. More general, a gene y is a direct descendant of some gene z if any other ancestor of y is also an ancestor of z. Moreover, a gene y is called a *common ancestor* of two genes x_1 and x_2 if y is an ancestor of both x_1 and x_2. In particular, y is the *last common ancestor* of x_1 and x_2, denoted by $\mathrm{lca}(x_1, x_2)$, if any other common ancestor of x_1 and x_2 is likewise an ancestor of y. The term of the last common ancestor is sometimes also denoted as *most recent common ancestor* or *lowest common ancestor* in the literature.

Similarly to gene trees, the leaves of species trees represent extant species while inner nodes refer to speciation events, i.e., the divergence of an ancestral species into two or more descendant species. The length of an edge from some node x to its descendant y is often interpreted as the life time of the ancestral species x. In this work, however, we are not primarily interested in specific branch lengths but mainly restrict our attention to tree topologies.
Furthermore, this thesis is mainly concerned with the reconstruction of gene trees.

A group of genes is called *homologous* if they all share the same ancestry. The genomic sequences of such genes are often similar to some extent. However, neither sequence similarity nor functional similarity shall be confused with homology since both can also be the result of convergent evolution, i.e., the independent evolution of similar traits of lineages that have separate evolutionary origins. In order to account for those differences, Walter M. Fitch in 1970 distinguished between *homology* and *analogy* [64]. He further refined the concept of homology by subdividing it into *orthology* and *paralogy* [64]:

> *"Where the homology is the result of gene duplication so that both copies have descended side by side during the history of an organism, (for example, alpha and beta hemoglobin) the genes should be called paralogous (para = in parallel). Where the homology is the result of speciation so that the history of the gene reflects the history of the species (for example alpha hemoglobin in man and mouse) the genes should be called orthologous (ortho = exact)."*

Although this distinction between orthology and paralogy in terms of evolutionary history is clear and concise, this terminology as well as its proper usage and its relation to gene function have been intensely debated at the turn of the millennium [184, 130, 76]. Over the last two decades, however, a strict phylogenetic definition seems to have prevailed: Homology, orthology, and paralogy are clearly and precisely defined by the common evolutionary history of a set of genes. In a collection of genomes, a *gene family* consists of all genes that have evolved from one common ancestor.
This work uses Fitch's definition, more precisely:

Definition 2.1. *Two genes x and y are called* homologs *if they have evolved from a common ancestral gene.*
They are called orthologs *if they have evolved from their last common ancestor by a speciation event, and* paralogs *if they have evolved from their last common ancestor by a duplication event.*

In the context of event-labeled gene trees, this means that the last common ancestor of two orthologous or paralogous genes is labeled as a "speciation" or "duplication", respectively. Fig. 1 shows an example of a gene family comprising the genes a_1, a_2, a_3, b_1, b_2, and c from three distinct species A, B, and C. Gene c from species C is, for instance, orthologous to any other gene in this gene family. Gene a_1 from species A, on the other hand, is orthologous to gene b_2 but not to b_1 from species B. The pair of genes a_1 and b_1 as well as the pair a_1 and a_2 are paralogs.

Note that homology as well as orthology and paralogy are symmetric and irreflexive relations. In contrast to homology, however, which is clearly a transitive relation, orthology and paralogy are non-transitive. In Fig. 1, for instance, a_1 and b_2 are both orthologous to c but paralogous to each other. In this work, orthology and paralogy will therefore be considered as binary relations.

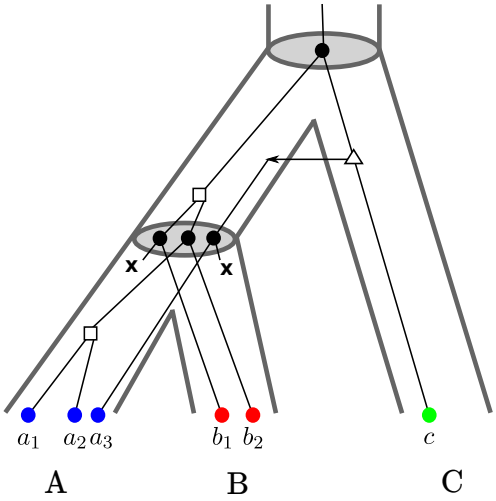


Fig. 1. The evolutionary history of a gene family comprising genes of the three species A, B, and C. The event-labeled gene tree is embedded in the tube-like species tree. The gene tree contains speciations (●), duplications (□), HGT events (△), and gene losses (×).

In addition, Sonnhammer and Koonin [210] further distinguish between paralogous genes emerging from lineage-specific or ancestral duplications (with respect to a specific speciation event), which they call *in-paralogs* and *out-paralogs*, respectively. In our example, a_1 and b_2 are out-paralogs w.r.t. the speciation event separating the species A and B and in-paralogs w.r.t. the speciation event separating C from A and B. Moreover, Sonnhammer and Koonin [210] denote two in-paralogs that are orthologous to the same gene as *co-orthologs*. Genes a_1 and a_2 are, for instance, co-orthologs of gene c in Fig. 1.

Although the notion of homology avoids any reference to functional similarity of genes, orthologous genes usually have equivalent functions in the corresponding organisms [131]. More strictly, the relationship of orthology and gene function is asymmetric, i.e., one cannot deduce orthology from similar function while the reverse implication is in most cases true. One-to-one orthologs, in particular, show functional equivalence more often than not [204]. In contrast, the functions of two paralogous genes are, albeit often related, clearly distinct from each other [115, 8, 214] (see also [167]). This is explained by the fact that redundant copies are not stable under mutational pressure. They need to diverge in function either by subfunctionalization or neofunctionalization [68], otherwise one copy will rapidly become dysfunctional and eventually be erased. Gene duplication, which may be caused for instance by unequal crossover or whole chromosome/genome duplication during reproduction, is known as a major force for evolution [230, 186, 176].

In addition to speciation and duplication events, the history of a gene family may also involve *horizontal* (also called *lateral*) *gene transfer (HGT)*, which is assumed to play an important role not only in prokaryotic [132] but also in eukaryotic evolution [125]. During an HGT event, genetic material is transferred between different species by means other than the "vertical" transmission from parent to offspring during reproduction. The result is a horizontally transferred *copy* and an *original* that continues to be vertically transferred. In contrast to speciation and gene duplication, horizontal transfer is thus inherently asymmetric. Horizontal gene transfer is captured by the concept of *xenology*. In contrast to orthology and paralogy, the formal definition of xenology is much less consistent in the biological literature. The most commonly used definition

was introduced by Walter M. Fitch in 2000 who calls two genes *xenologs* if at least one horizontal gene transfer event occured along the evolutionary history since their last common ancestor [66, 120]. More formally:

Definition 2.2. *A gene x is called* xenologous *to gene y if genetic material has been horizontally transferred between species along the evolutionary history of x since the last common ancestor of x and y. The genes x and y are called* xenologs *if either x is xenologous to y or y is xenologous to x.*

In Fig. 1, for instance, gene a_3 is xenologous to any other gene in this gene family.

More often then not, trees that have been inferred from sets of genes consisting of both orthologs and paralogs are inconsistent with the true species tree. Since the evolution of orthologs is at least roughly clock-like, i.e., the evolution rate is at least approximately constant [204], and is thus assumed to reflect the evolution of the species tree, molecular phylogenetics strives to exclude paralogs from the analysis and exclusively restrict its attention to one-to-one orthologs. Correct orthology assignment lies at the heart of genome annotation, functional annotation and gene nomenclature. Moreover, orthologs are often used as anchors for chromosome alignments, which form the basis for synteny-based methods [209], such as *forward genomics*, a computational strategy that identifies gene loss by associating specific genomic regions with lost phenotypes [107]. A high-quality data set of orthologs is also an important prerequisite for the reconstruction of ancestral proteomes.

In the presence of HGT events, different parts of the genome have distinct evolutionary histories which significantly complicates the inference of evolutionary relatedness in many cases. Several methods have been devised that use sequence features to detect HGT events [142, 50, 190, 188] (see also Section 2.6). However, their is a strong tendency for different methods to infer different HGT events.

One of the major problems in orthology detection is the identification of orthologous genes among a set of genes comprising all sorts of homologous genes, including paralogs and xenologs. A huge variety of orthology inference methods has been developed, which mainly fall into two distinct classes: tree-based and graph-based methods (see [134] for a detailed survey). This work is concernced with a new graph-based approach.

2.2 THE MATHEMATICS OF ORTHOLOGY

Somewhat surprisingly, orthology, paralogy, and xenology have only recently moved into the focus of a systematic formal, i.e., mathematical, analysis. As a consequence, the problem of estimating orthology directly from (dis-)similarity data is still not completely understood, although several research groups have made significant progress (see e.g. [105, 97, 98, 74]). The seminal work on "symmetric dating maps" and "symbolic ultrametrics" [21] served as the starting point.

Orthology and paralogy are uniquely determined by (i) a gene tree reflecting the evolutionary history of the gene family under consideration, (ii) the

species tree of the corresponding set of species, and (iii) the mutual relation, i.e., the reconciliation, of these two trees. The reconciliation maps the vertices of the gene tree to the vertices and edges of the species tree by preserving the ancestor relation of the gene tree. In particular, vertices corresponding to a speciation event in the gene tree are associated with inner vertices of the species tree, whereas vertices labeled by "duplication" of "horizontal gene transfer" are mapped to the edges of the species tree. More formally, let $T = (V(T), E(T))$ with leaf set $L(T)$ and root 0_T, and $S = (V(S), E(S))$ with leaf set $L(S)$ and root 0_S be the gene and species tree, respectively, where both trees are planted phylogenetic trees, i.e., their roots have degree 1 and any inner vertex has degree at least 3. Moreover, the ancestor order of a tree T is given by $x \prec_T y$ whenever the vertex y lies on the unique path from x to the root of T, where $x \preceq_T y$ if $x = y$ or $x \prec_T y$. Then, for a given surjective map $\sigma : L(T) \to L(S)$ that assigns to each gene the species in which it resides, the reconciliation from (T, σ) to S is a map $\mu : V(T) \to V(S) \cup E(S)$ satisfying the following, natural axioms [74]:

(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.* $x \prec_T y$ implies $\mu(x) \preceq_S \mu(y)$.

(R3) *Speciation Constraints.* Suppose $\mu(x) \in V(S) \setminus L(S)$.

- (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.

The embedding of a gene tree into a species tree in Fig. 1 serves as an example of a reconciliation map. Note that there typically exist more than one possible reconciliation for a given pair of gene and species trees. As will be shown later in Section 6, the axiom system above is equivalent to another version, which has been commonly used in the literature (see e.g. Górecki and Tiuryn [79], Vernot et al. [227], Doyon et al. [57], Rusin et al. [194], Hellmuth [91], Nøjgaard et al. [173], and the references therein). An extension of these axioms to reconciliation scenarios containing HGT events can be found e.g. in [224, 18, 173, 74].
Key results obtained in this setting include the following:

- The orthology relation has the structure of a cograph [97, 95].
- A cograph is uniquely represented by its cotree T, that is, a phylogenetic tree whose interior nodes are labeled by the type of event (speciation, duplication). Hellmuth et al. [97] showed that the cotree is always a homeomorphic image (i.e., a not necessarily fully resolved version) of the gene tree.
- The cograph property remains intact in the presence of horizontal transfer provided xenologous pairs are treated as neither orthologs nor paralogous [100], i.e., there exists an event-labeled gene tree for an estimated event-labeled relation if and only if this relation is a directed cograph.

- The (possibly incompletely resolved) event-labeled gene trees also imply, in the absence of HGT, a complete characterization of the possible species trees with which they can be reconciled [105, 91].

- In the presence of HGT, event-labeled gene trees imply at least necessary conditions on the structure of possible species trees [91]. With HGT, time-consistency must be ensured, i.e., no sequence of HGT events can give raise to a directed cycle in the reconciled tree. First attempts to handle this issue, including some timing constraints, can be found in [31, 161, 57, 224]. A first characterization and algorithm for determining whether a given pair of event-labeled gene and species tree can be reconciled in a time-consistent way has been provided in [173].

It is important to note that there is a crucial distinction between unlabeled and event-labeled gene trees: In the unlabeled case, the so-called Last Common Ancestor (LCA) map, which maps any node u of the gene tree T to the last common ancestral species of all its descendant genes, i.e., $\lambda(u) := \mathrm{lca}_S(\sigma(L(T(u))))$, is always a valid reconciliation between any gene tree T and any species tree S, as long as $\sigma(L(T)) = L(S)$. The LCA map can be computed in linear time [239]. In the event-labeled case, however, S must display the set of species triples that is implied by gene triples from three different species with a speciation event as their last common ancestor [105]. Hellmuth et al. [98] demonstrated that the constraints on the species tree are sufficiently strong to infer a fully resolved phylogeny, e.g. the phylogeny of the Aquificiales from an estimate of the orthology relations for the individual gene families provided by `ProteinOrtho` [144].

2.3 DIRECT INFERENCE OF THE RECONCILIATION MAP

The reconciliation map and the event labeling do constrain each other. Since the distinction of orthology and paralogy is defined in terms of the event labeling, we see that orthology explicitly depends on the choice of the corresponding reconciliation map. One can therefore rephrase the problem of orthology assignment as the task of approximating the true reconciliation map μ^*. In this context, a systematic exploration of the space of possible reconciliations [55] is of interest. A classical approach to estimate μ^* starts from the observation that the discordance between gene and species trees are complex histories of gene duplications [179], typically combined with subsequent gene losses. It is natural then to search for a parsimonious explanation. This requires a cost model for scoring the reconciliation maps. The most widely used cost models are the duplication cost model, pioneered by Goodman et al. [77] and Page [178], and the duplication-loss-cost (or mutation cost) model, first introduced by Guigo et al. [85], that minimize the number of duplications, resp., the number of duplications and losses. While these costs can be computed in linear time using the LCA mapping for a given pair of gene and species tree, the problem of finding a species tree for a given family of gene trees is shown to be NP-hard in both models [153]. There exist, however, local search heuristics [17, 229] and more recently, even exact dynamic programming solutions [30, 217] have

been developed. Moreover, variations of this model have been analyzed that treat different types of input trees such as unrooted trees [81], non-binary trees [140, 240], or even erroneous trees [78]. Despite its importance in practice, not much is known about the mathematical properties of duplication cost models, but see [82] for recent advances. As an alternative, the loss cost model, which penalizes the number of inferred gene loss events, has been proposed in [32]. Yet another scoring scheme assumes that the incongruence between gene and species tree is caused by incomplete lineage sorting (see for example [154]), which implies the so-called reconciliation cost model. Complementing the parsimony-based reconciliation methods, there exist also fully probabilistic models of reconciliation using essentially equivalent cost models [13, 14, 80].

2.4 Inference of the orthology relation

There exist two fundamentally different approaches ("tree-based" and "graph-based") to estimate orthology from genomic sequence data.

The tree-based approach starts with a pair of gene and species tree and then computes a reconciliation together with a corresponding event labeling, from which an orthology relation can be inferred [204]. This indirect method serves as basis for many algorithms and software tools [133, 207, 112, 1, 113]. Despite being often considered as more accurate than graph-based methods [134], tree-based methods suffer from all the difficulties of large-scale phylogenetic inference, such as high computational complexity, strong dependency on the accuracy of underlying multiple sequence alignments [222], and sensitivity to noise in the data due to, for instance, long-branch attraction [20, 175]. Moreover, many tree-based approaches depend on species trees, thus knowledge about one-to-one orthologs to reconstruct the correct species tree is crucial for the accuracy of those methods.

Apart from parsimonious reconciliation models, more recently developed probabilistic models try to co-estimate gene tree, species tree, and reconciliation maps [13, 4, 189, 238, 165], some of them also including horizontal gene transfer [208]. However, these fully probabilistic approaches are computationally highly complex, limiting their applicability to very large datasets.

The second approach is the one we are concerned with in this thesis. So-called graph-based methods infer orthology relations directly from sequence data without constructing gene or species trees in advance. Examples include `COG Database` [221], `eggNOG` [119], `OrthoMCL-DB` [33], `OMA Browser` [199], `InParanoid` [19], `OrthoDB` [135], and `KEGG` [123], and `ProteinOrtho/POFF` [144, 145]. Their common starting point are reciprocal best matches, also known as symmetric best matches [220], bidirectional best hits (BBH) [177], reciprocal best hits (RBH) [23], or reciprocal smallest distance (RSD) [228]. Usually estimated by a fast method for sequence comparison, the basic idea is to capture pairs of genes a and b from species A and B, respectively, that are evolutionarily most closely related. Since orthologs derive from a speciation event, it is necessary for a and b to be orthologs that a is the closest relative of b in A, and vice versa. We will return to this key point in more detail in Section 2.5.

Many orthology detection methods are based on this simple idea and mainly differ in two important aspects: (i) best matches can be determined using different (dis-)similarity measures (see Section 2.5 for further details), and (ii) reciprocal best matches can be assumed to be one-to-one or many-to-many. Many of those methods summarize orthologous genes as *clusters of orthologous groups* (COGs) [220]. However, as already noted in Section 2.1, the orthology relation is in general non-transitive, thus COGs can only serve as an approximation for the true orthology relation.

Within the original version of the `OMA` method [48], COGs are exclusively extracted from stable pairs, that is estimated one-to-one orthologs, which are represented as a graph G_{1-1}. Since the one-to-one orthology relation is, by construction, transitive, orthologous groups are expected to form cliques, that is complete subgraphs, in G_{1-1}. The true orthology relation thus contains the COGs from `OMA`. Those are identified by first looking for maximal cliques and then partitioning G_{1-1} into vertex disjoint cliques such that the number of edges within these cliques is maximal (minimum edge clique partition problem) [193, 204]. Both problems are in general NP-hard but become very easy if the input graph corresponds to a mathematically correct orthology relation, i.e., a cograph [71].

In other frameworks, including more recent versions of `OMA` [193], reciprocal best matches are allowed to form many-to-many relations, thus orthologs and paralogs can be included within the same COG [220]. The initial starting point is a similarity graph built from all-against-all sequence comparisons that contains an edge between two genes if and only if their sequence similarity exceeds a pre-defined threshold. COGs are then extracted from this similarity graph by a wide variety of clustering techniques. The resulting COGs heavily depend on parameters which determine the trade-off between stringency and size of those clusters. A wide variety of clustering strategies has been developed to reconstruct COGs, which constitutes the main difference between the available software tools. [204]

Several tools, such as `InParanoid` [191, 19] or `OrthoMCL-DB` [33], separately identify recent in-paralogs by assuming that the distance between them must be smaller than the distance between their corresponding species. Various types of filters, such as synteny information in `PoFF` [145], or an extended clustering step to detect in-paralogs as used in `InParanoid`, are applied to reduce potential false positives.

While most of the older tools focused on groups of co-orthologs and conceptually treated orthology identification as clustering algorithms (e.g. `COG Database`, `eggNOG`, `OMA`), more recent implementations mostly focus on orthology as a binary relation. These tools typically employ additional filters to detect false positives among the initial orthology assignments. An example is a "witnesses of paralogy", that is, a third species in which both paralogs have survived to resolve cases as in Fig. 2 [49]: If c_1 and c_2 are paralogs in species C and the quartet $ac_1|bc_2$ is the only quartet that can be inferred from the additive evolutionary distances, i.e., $d(a,c_1)+d(b,c_2) \ll d(a,c_2)+d(b,c_1), d(a,b)+d(c_1,c_2)$ for some candidate orthologs a in A and $b \in B$, then a and b must be paralogs. Hence, the initially inserted edge ab corresponds to a false positive orthology as-

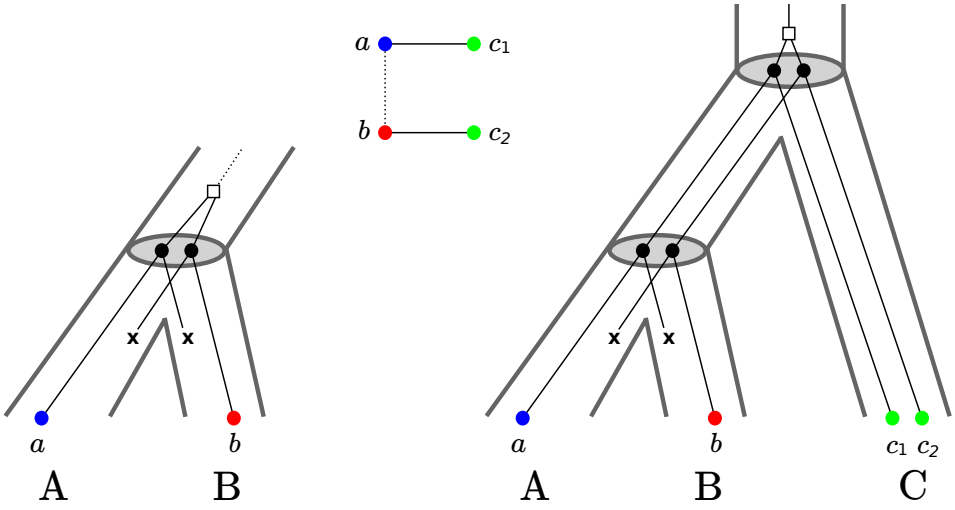


Fig. 2. *Left:* Scenario of an undetectable paralogy, where only one paralog of an ancient duplication survived in species A and B, respectively. By only comparing species A and B, there is no evidence for duplication or gene loss events, hence reciprocal best hit methods erroneously identify the genes a of species A and b of species B as potential orthologs.
Right: Both paralogs of the ancient duplication survived in a third species C. Including C in the analysis yields an orthology relation that is not a cograph and must thus be edited. In this case, the induced P_4 (shown in the middle) can be edited to the cograph $K_2 \cup K_2$ by deletion of the false positive orthology edge between a and b.

signment and can thus be removed [204]. One of the first tools which is based on this quartet-based approach for distinguishing between orthologs and paralogs is `QuartetS-DB` [236, 237]. A similar approach for evaluating and improving orthology predictions that relies on phylogenetic distance ratios between three species, is addressed by the tool `Ortholuge` [69, 231]. Reviews and benchmarks of different graph-based orthology detection methods support the efficiency of best match based approaches [134, 169, 6]. Recently, a web-based service for standardized orthology benchmarking has been developed [10].

As a consequence of non-clock-like rates of evolution, discrepancies between specific sequence-based distance measurements and evolutionary divergence times, and noise in the data, both false positive and false negative orthology assignments are unavoidable in practice [144, 7, 9, 139, 52, 141, 53, 54]. Empirically determined orthology relations thus usually violate the cograph property. Cographs, however, are a highly restricted class of graphs. Enforcing the cograph property therefore provides a very efficient means of noise reduction [98]. The cograph editing problem has been shown to be NP-hard [149, 150], however it remains tractable if the estimated orthology relation is not far from the true orthology relation. The tool `ParaPhylo` uses ILP-based cograph editing, which corrects errors in the estimates by looking for tree-representable sets of relations that are closest to the estimates. This approach has been shown to sufficiently reduce noise in order to find event-labeled gene trees, even with high levels of HGT in the data [98].

The co-orthologs in a species B of a given gene a from species A are by definition (ignoring horizontal gene transfer) the closest relatives of a in B. Therefore, reciprocal best matches constitute a natural approximation of the concept of reciprocal evolutionary closest relatedness of genes. Evolutionary relatedness is an inherently phylogenetic property. It is therefore most naturally defined relative to a gene tree T, more precisely gene b from species B is a best match of a from species A if and only if their last common ancestor is a descendant of the last common ancestor of b and any other gene from species A. If a is also a best match of b, then a and b are reciprocal best matches. Best matches can thus be equivalently expressed in terms of evolutionary (divergence) time. Divergence times and genetic distances are only equivalent under the assumption of a molecular clock [241, 137]. Although the strong assumption that the molecular clock is at least reasonably approximated, is frequently violated in real-life data, best match heuristics still perform surprisingly well in the context of orthology prediction [233]. In fact, reciprocal best match heuristics perform at least as good for this task as methods that first estimate the gene phylogeny [10, 204], although there are practical problems, in particular in applications to eukaryotic genes [41].

The main reason for the resilience of RBH methods is that the identification of best matches only requires inequalities between sequence similarities. In particular, therefore they are invariant under monotonic transformations of the distance measures, and, in contrast e.g. to distance based phylogenetic methods, do not require additivity. Even more generally, it suffices that the evolutionary rates of the different members of a gene family are roughly the same within each lineage.

Nevertheless, RBH methods are far from perfect. Large differences in evolutionary rates between paralogs, as predicted by the DDC model [68], for example, may lead to false negatives among co-orthologs and false positive best matches between members of slowly evolving subfamilies. Recent orthology detection methods recognize the sources of error and complement sequence similarity by additional sources of information. Most notably, synteny is often used to support or reject reciprocal best matches [145, 116]. Another class of approaches combines the information of small sets of pairwise best matches to improve orthology prediction [236, 226].

In order to identify reciprocal best matches, RBH methods typically first compute and rank for each gene of species A the sequence similarity with every gene of species B in decreasing order, from which reciprocal best matches are then extracted. The pairs of genes that are identified as reciprocal best matches, however, heavily depend on the chosen method for measuring sequence similarity. Because of its lower computational complexity, many methods use `BLAST` or `BLAT` scores for the derivation of reciprocal best matches (e.g. `InParanoid` [191, 19], `OrthoMCL` [33]). However, these methods differ in the thresholds used for best match assignment, depending for instance on whether one-to-one or many-to-many best hits are considered, as well as in the exact choice of parameters used for the `BLAST` search. An investigation of different `BLAST` parameters

and comparison of reciprocal best hits to reciprocal smallest distances can be found in [163]. As lower BLAST scores for short sequences are thought to cause many short sequences not to be assigned to an orthologous group at all, the more recently developed tool OrthoFinder [63] proposes a modified BLAST score that derives sequence similarity from BLAST bit scores by taking gene length into account in order to reduce the impact of gene length on the clustering accuracy.

Wall et al. [228] proposed reciprocal smallest distances, which are used for instance in the tool Roundup [46], as a more accurate estimate of evolutionary distances. To this end, exact pairwise alignments are generated for the top BLAST hits from a query sequence from species A, from which the maximum likelihood distances between A and the top BLAST hits are estimated in order to identify the gene with the smallest evolutionary distance from A. These estimated distances are thought to perform better than BLAST hits, however simulation results in [163] suggest that this is not necessarily true.

Another approach is implemented in Orthograph [183] which retrieves reciprocal best matches from a profile Hidden Markov Model based search. This tool builds upon the method HaMStR [59].

Somewhat surprisingly, this evolutionary notion of best matches has received very little attention in the literature, despite the wide-spread use of pragmatic RBH heuristics in computational biology. Chapters 4, 5, and 6 thus systematically investigate the relationships between (reciprocal) best matches and the underlying gene tree. These results will be further used in Chapter 6 to successfully identify a considerable fraction of false positive orthology assignments among reciprocal best hits.

2.6 XENOLOGY

Methods for inferring horizontal gene transfer are less well established and investigated than orthology inference methods. Current approaches for inferring HGT events are either based on evolutionary history ("phylogenetic"), which can be explicit or implicit, or on sequence composition ("parametric") [190, 16].

Parametric methods use genomic signatures, such as nucleotide composition [143] or codon usage bias [164, 160], and identify genomic regions deviating from that specific signature as HGT events [44]. These methods rely on the fact that the genomic properties of the transferred gene are still those of the donor genome, which makes the transferred gene distinguishable from the acceptor genome. For this reason, however, the performance of parametric methods highly depends on the amount of (dis-)similarity in the evolutionary patterns of the organisms in question. Moreover, transferred sequences adapt quite rapidly to their new host genome, hence parametric methods are limited to recent HGT events. [50]

Explicit phylogenetic methods identify HGT by looking for genes that are involved in conflicts between gene trees and a reference species tree [203] and therefore heavily depend on the accuracy of the gene and species trees, whose reliability is often low [7]. A first attempt to infer horizontal gene transfer events directly from sequence data are implicit phylogenetic approaches, such as

DLIGHT [50]. Those methods compare sequence distances, where unexpectedly short or long distances are suggested to correspond to HGT events.

The mathematics of xenology has attracted very little attention so far. Only very recently, a first mathematical characterization of xenology in terms of a binary relation has been explored by Hellmuth et al. [100]. This "lca-xenology" relation uses directed cographs to capture the directional aspect of HGT, however, it remains unclear to what extent it can be directly inferred from sequence similarity data [100].
A formalization of Fitch's xenology concept is presented in Chapter 8 in the form of a not necessarily symmetric binary relation.

3

BASIC DEFINITIONS

3.1 SETS AND BINARY RELATIONS

Throughout this work, all sets of elements are always assumed to be finite. Given two sets V and W, we write $W \subseteq V$ ($W \subset V$) for W being a (proper) subset of V. Moreover, $V \cup W$, $V \uplus W$, $V \cap W$, $V \triangle W$, and $V \setminus W$ denote the union, the disjoint union, the intersection, the symmetric difference, and the set difference, resp., of V and W. A *partition* of a set V is a collection of disjoint non-empty sets $V_1, ..., V_k$, $k \geq 1$, such that $V = V_1 \uplus ... \uplus V_k$. Two sets V and W *do not overlap* if $V \cap W \in \{\emptyset, V, W\}$, and they *overlap*, otherwise. The set of unordered pairs of elements from V is defined by $\binom{V}{2} := \{\{x, y\} \mid x, y \in V, x \neq y\}$, whereas the respective set of ordered pairs is given by $V \times V := \{(x, y) \mid x, y \in V\}$. The set $V_\times^{irr} := \{(x, y) \mid x, y \in V, x \neq y\}$ is the irreflexive part of $V \times V$. The *power set* of V is the set of all subsets of V, denoted by 2^V.

A set $\mathcal{B} \subseteq V \times V$ of ordered pairs (x, y) with $x, y \in V$ is called a *binary relation*. Instead of $(x, y) \in \mathcal{B}$, we will often write $x\mathcal{B}y$. Throughout this work, all relations are binary. If not stated otherwise, we therefore simply speak of a "relation" without explicitly mentioning "binary". The binary relation $\mathcal{B}$ is *irreflexive* if $x \neq y$ whenever $x\mathcal{B}y$, and *reflexive* otherwise. Moreover, a relation $\mathcal{B}$ satisfying $x\mathcal{B}y$ if and only if $y\mathcal{B}x$ for any pair $x, y \in V$, is called *symmetric*. Furthermore, the relation $\mathcal{B}$ is called *transitive* if $x\mathcal{B}y$ and $y\mathcal{B}z$ implies $x\mathcal{B}z$. A binary relation $\mathcal{B}$ on V that is reflexive, symmetric, and transitive is called *equivalence relation*. The *equivalence class* of an element $x \in V$ is the set $[x] := \{y \in V \mid x\mathcal{B}y\}$.
Furthermore, given a binary relation $\mathcal{B}$, the set V is a *partially ordered set* or *poset* if, for all $x, y, z \in V$, it holds (i) $x\mathcal{B}x$ (reflexivity), (ii) $x\mathcal{B}y$ and $y\mathcal{B}z$ implies $x\mathcal{B}z$ (transitivity), and (iii) $x\mathcal{B}y$ and $y\mathcal{B}x$ implies $x = y$ (anti-symmetry).

3.2 GRAPHS

A *graph* $G = (V, E)$ is an ordered pair of sets V and E, where V is a set of *vertices* (or *nodes*) and E a set of *edges* (or *arcs*). The vertex and edge set of G are also often denoted as $V(G)$ and $E(G)$, resp., whenever the reference graph is not obvious. The graph G is called *undirected* if $E \subseteq \binom{V}{2}$ and *directed* (or *digraph*) if $E \subseteq V_\times^{irr}$. This definition of a graph explicitly excludes multiple edges between the same vertices as well as loops, i.e., edges connecting a vertex with itself. Such graphs are usually called *simple* in the literature. Throughout this work, $G = (V, E)$ and $\vec{G} = (V, \vec{E})$ denote simple undirected and simple directed graphs, respectively. Moreover, directed arcs (x, y) in a digraph $\vec{G}$ will be distinguished from edges xy in an undirected graph G or tree T (see Section 3.3 for a definition of trees).

Two undirected graphs G and G' are called *isomorphic* if there exists a bijection $\phi : V(G) \to V(G')$ such that, for any two vertices $x, y \in V(G)$, it holds $xy \in E(G)$ if and only if $\phi(x)\phi(y) \in E(G')$. The corresponding definition for directed graphs is obtained by simply substituting in the definition for undirected graphs the edges xy and $\phi(x)\phi(y)$ by (x, y) and $(\phi(x), \phi(y))$, respectively. For the purpose of this work it will not be relevant to distinguish two isomorphic graphs.
Throughout this work, relations are represented as graphs. The terms graph and relation are therefore used interchangeably.

3.2.1 *Adjacency, Neighborhoods, and Paths*

Two vertices of an undirected or directed graph are called *incident* or *adjacent* if they are connected by an edge/arc. The *neighborhood* $N(x)$ of some vertex x in a given graph is the set of all its adjacent vertices, the *neighbors* of x. The number of neighbors of x is called the *degree* of x, denoted by $\deg(x)$. For a vertex $x \in V$ of a digraph $\vec{G} = (V, \vec{E})$, the out- and in-neighborhood is denoted by $N^+(x) \coloneqq \{y \in V \mid (x, y) \in \vec{E}\}$ and $N^-(x) \coloneqq \{y \in V \mid (y, x) \in \vec{E}\}$, respectively. This notation naturally extends to sets of vertices $A \subseteq V$: $N^+(A) = \bigcup_{x \in A} N^+(x)$ and $N^-(A) = \bigcup_{x \in A} N^-(x)$. The number of out- and in-neighbors, resp., of x in $\vec{G}$ is the *out-degree*, denoted by $\deg^+(x)$, and *in-degree*, denoted by $\deg^-(x)$, of x.

A sequence of vertices $S = (x_1, ..., x_n)$ in an undirected graph $G = (V, E)$ is called a *path* if $x_i x_{i+1} \in E$ for any $1 \leq i \leq n-1$ and all vertices of S are pairwise distinct. Moreover, S is a *cycle* if it is a path and $x_n x_1 \in E$. Similarly, a sequence of vertices $S = (x_1, ..., x_n)$ in a digraph $\vec{G} = (V, \vec{E})$ is called a *path* if $(x_i, x_{i+1}) \in \vec{E}$ for any $1 \leq i \leq n-1$ and all vertices of S are pairwise distinct. Moreover, S is a *cycle* if it is a path and $(x_n, x_1) \in \vec{E}$.

3.2.2 *Subgraphs and Connectedness*

A graph $G' = (V', E')$ is a *subgraph* of $G = (V, E)$, denoted by $G' \subseteq G$, where both graphs may be undirected or not, if $V' \subseteq V$ and $E' \subseteq E$. If in addition, in case of undirected graphs, $xy \in E$ implies $xy \in E'$ for any $x, y \in V'$ or, in case of digraphs, $(x, y) \in E$ implies $(x, y) \in E'$ for any $x, y \in V'$, then the subgraph G' is called *induced subgraph of* G, denoted by $G' \coloneqq G[V']$. By abuse of notation, we will often write $G[x, y, z]$ instead of $G[\{x, y, z\}]$ for $x, y, z \in V$.

An undirected graph $G = (V, E)$ is *connected* if for any two distinct vertices $x, y \in V$ there exists a path connecting x and y. Similarly, a digraph is called *connected* in this work whenever its underlying undirected graph (obtained by ignoring the direction of the arcs) is connected. A *connected component* of an undirected or directed graph G is a maximal connected subgraph of G, and G is *disconnected* if it contains more than one connected component. Moreover, a digraph $\vec{G} = (V, \vec{E})$ is *strongly connected* if, for any two distinct vertices x and y in V, it contains a path from x to y. A maximal strongly connected subgraph of $\vec{G}$ is called a *strong connected component*.

3.2.3 *Graph Operations*

The complement $\overline{G} = (V, \overline{E})$ of a graph $G = (V, E)$ has edge set $\overline{E} = \binom{V}{2} \setminus E$ if G is undirected and $\overline{E} = (V \times V) \setminus E$ if G is directed. The join of two undirected disjoint graphs $G = (V, E)$ and $G' = (V', E')$ is defined by $G \triangledown G' = (V \cup V', E \cup E' \cup \{xy \mid x \in V, y \in V'\})$, whereas their disjoint union is given by $G \uplus G' = (V \cup V', E \cup E')$. These definitions can be naturally extended to directed graphs: For two directed disjoint graphs $\vec{G} = (V, \vec{E})$ and $\vec{G'} = (V', \vec{E'})$, their join is given by $\vec{G} \triangledown \vec{G'} = (V \cup V', \vec{E} \cup \vec{E'} \cup \{(x, y), (y, x) \mid x \in V, y \in V'\})$ and their disjoint union by $\vec{G} \uplus \vec{G'} = (V \cup V', \vec{E} \cup \vec{E'})$. Moreover, the *order composition* of $\vec{G}$ and $\vec{G'}$ is the disjoint union of the two graphs plus all arcs (x, y) with $x \in V(\vec{G})$ and $y \in V(\vec{G'})$.

3.2.4 *Special Graphs*

A graph consisting of a single isolated vertex without edges is called *singleton*. An undirected graph with n vertices is called *complete*, denoted by K_n, if any two distinct vertices are adjacent in K_n. Similarly, a directed graph $\vec{G} = (V, \vec{E})$ is *complete* if (x, y) and (y, x) are both edges in $\vec{G}$ for any pair of vertices $x, y \in V$. A graph with vertex set V, undirected or not, is called *k-partite* if its vertex set can be partitioned into $k \geq 2$ pairwise disjoint *independent sets* $V_1, ..., V_k$, i.e., $x \in V_i$ and $y \in V_j$ with $i \neq j$ for any two vertices $x, y \in V$ that are connected by an edge. The graph is called *bipartite* for $k = 2$ and *multipartite* otherwise. Moreover, an undirected graph G is called *complete multipartite*, denoted by $K_{n_1,...,n_k}$, if it is k-partite and $xy \in E(K_{n_1,...,n_k})$ for any two distinct vertices x, y satisfying $x \in V_i, y \in V_j$ with $i \neq j$, where $V(K_{n_1,...,n_k}) = V_1 \cup ... \cup V_k$. If G is directed, then it is *complete multipartite* if it is k-partite and $(x, y), (y, x) \in E(G)$ for any $x \in V_i, y \in V_j$ with $i \neq j$.

3.2.5 *Vertex-Colored graphs*

A graph $G = (V, E)$, undirected or not, is *vertex-colored* (or simply *colored*) if there is a non-empty set of colors S and a map $\sigma : V \to S$ that assigns a color to each vertex. Such a graph is denoted by (G, σ). A vertex coloring σ is *proper* if $\sigma(x) \neq \sigma(y)$ for any two adjacent vertices $x, y \in V(G)$. We will furthermore assume throughout this work that the map $\sigma : L \to S$ is surjective. For a subset $L' \subseteq L$ we write $\sigma(L') = \{\sigma(x) \mid x \in L'\}$. Moreover, we use the notation $\sigma_{|L'}$ for the surjective map $\sigma : L' \to \sigma(L')$. In particular, for $V' \subseteq V$, the colored induced subgraph of G, whose coloring is obtained by restricting σ to V', is denoted by $(G, \sigma)[V'] \coloneqq (G[V'], \sigma_{|V'})$. We write $V[s] = \{x \in V \mid \sigma(x) = s\}$ for the set of all vertices with color s in (G, σ). In particular, for given colors $r, s, t \in S$, we write $(G_{st}, \sigma_{st}) \coloneqq (G[V[s] \cup V[t]], \sigma_{|V[s] \cup V[t]})$ and $(G_{rst}, \sigma_{rst}) \coloneqq (G[V[r] \cup V[s] \cup V[t]], \sigma_{|V[r] \cup V[s] \cup V[t]})$ for the respective induced subgraphs.

The neighborhood of some vertex $x \in V$ restricted to a color $s \in S$ will be denoted by $N_s(x) \coloneqq \{z \mid z \in N(x) \text{ and } \sigma(z) = s\}$. Similarly, we write $N_s^+(x) \coloneqq$

$\{z \mid z \in N^+(x) \text{ and } \sigma(z) = s\}$ and $N_s^-(x) \coloneqq \{z \mid z \in N^-(x) \text{ and } \sigma(z) = s\}$ for the in- and out-neighborhoods of x with color s.

Moreover, for an undirected colored graph (G, σ), we write $\langle x_1 \dots x_k \rangle \in \mathcal{P}_k$ to denote that the vertices $x_1, \dots, x_k$ form an induced path $P = \langle x_1 \dots x_k \rangle$ on k vertices in G and with edges $x_i x_{i+1}$, $1 \leq i \leq k-1$. Analogously, $\langle x_1 \dots x_k \rangle \in \mathcal{C}_k$ denotes the fact that the vertices $x_1, \dots, x_k$ induce a cycle $C = \langle x_1 \dots x_k \rangle$ on k vertices with edges $x_i x_{i+1}$, $1 \leq i \leq k-1$, and $x_k x_1$. An induced cycle on six vertices is called *hexagon*. We will write that $\langle x_1 \dots x_k \rangle \in \mathcal{P}_k$, resp. $\mathcal{C}_k$ is of the form $(\sigma(x_1), \dots, \sigma(x_k))$ to indicate the vertex colors along induced paths, resp., cycles.

Two colored graphs (G, σ) and (G', σ') (undirected or not) are called *isomorphic* if G and G' are isomorphic and there exists a permutation $\pi : \sigma(V(G)) \to \sigma'(V(G'))$ of the colors. For the purpose of this work it will not be relevant to distinguish two colored graphs (G, σ) and (G', σ') that are isomorphic in the sense of isomorphic colored graphs, i.e., we do not distinguish permutations of colors.

3.3 TREES

A *tree* is a connected undirected graph without cycles. A *rooted tree* $T = (V, E)$ is a tree with one distinguished inner vertex $\rho_T \in V$ that is called the *root of* T. The *leaf set* $L \subseteq V$ (or $L(T)$ in case of ambiguity) consists of all vertices distinct from the root that have degree 1. Vertices in $V^0 \coloneqq V \setminus L$ (including ρ_T) are called *inner vertices*. A rooted tree $T = (V, E)$ on L is *phylogenetic* if its root has $\deg(\rho_T) \geq 2$ and every other inner vertex $v \in V^0 \setminus \{\rho_T\}$ has $\deg(v) \geq 3$. If the degree of each vertex $v \in V^0 \setminus \{\rho_T\}$ is exactly three and $\deg(\rho_T) = 2$, then the phylogenetic tree is called *binary*.
Throughout this work we are exclusively concerned with rooted phylogenetic trees unless explicitly stated otherwise.

Moreover, two rooted trees T and T' on the same leaf set L are *isomorphic* if there exists a bijection $\Phi : V(T) \to V(T')$ inducing a graph isomorphism from T to T', which maps the root of T to the root of T' and is the identity on L.
In this work we do not distinguish between isomorphic trees (unless explicitly stated). In particular, whenever speaking of "unique trees", this refers to "uniqueness up to isomorphism".

3.3.1 *Special Trees*

The star tree S_n is the complete multipartite graph $K_{1,n}$, i.e., the tree with exactly one internal node (the root) and n leaves.
Moreover, a rooted tree T is a *caterpillar* if every inner vertex has at most one child that is an inner vertex (see also [88]).
An *ordered* tree is a rooted tree with a specified ordering for the children of each vertex.

Furthermore, we extend the notion of a phylogenetic tree to so-called *planted (phylogenetic) trees*, which will be extensively used in Chapters 6 and 7. A planted phylogenetic tree is a rooted tree T with vertex set $V(T)$ and edge

set $E(T)$ such that (i) the root 0_T has degree 1 and (ii) all inner vertices have degree at least 3. We write $L(T)$ for the leaves (not including 0_T) and $V^0 = V(T) \setminus (L(T) \cup \{0_T\})$ for the inner vertices (also not including 0_T). The *conventional root* ρ_T of T is the unique neighbor of 0_T. It will sometimes be useful to consider $T(u)$ as planted tree by including the unique parent v of u and the edge vu.

3.3.2 *The Ancestor Relation and the Last Common Ancestor*

Given a rooted tree $T = (V, E)$, a vertex $u \in V$ is an *ancestor* of $v \in V$, in symbols $u \succeq_T v$, and v is a *descendant* of u, $v \preceq_T u$, if u lies on the unique path from v to the root ρ_T. We write $u \succ_T v$ ($v \prec_T u$) for $u \succeq_T v$ ($v \preceq_T u$) and $u \neq v$. If $v \preceq_T u$ or $v \succeq_T u$, then u and v are *comparable*, and *incomparable* otherwise. For a subset $A \subseteq V$ we write $A \preceq_T u$ to mean that $x \preceq_T u$ for all $x \in A$. If $uv \in E$ in T and $u \succ_T v$, we call u the *parent* of v, denoted by $\mathsf{par}_T(v)$, and define the *children* of u as $\mathsf{child}_T(u) := \{w \in V \mid uw \in E\}$. We denote two leaves $v, w \in L$ as *siblings* if $v, w \in \mathsf{child}_T(u)$ for some $u \in V$. The ancestor order is extended to edges by defining each edge $e = uv$ to be located between its incident vertices, i.e., $v \prec_T e \prec_T u$. In particular, by writing $e = uv$, we assume that u is closer to the root of T than v. Moreover, we say that e is an *outer* edge if $v \in L$ and an *inner* edge otherwise. In analogy to inner vertices, we refer to the set of inner edges of T as $E^0(T)$.
For $v \in V$, we denote by $T(v)$ the subtree rooted at v, that is the induced subgraph $T[V']$ with root v, where $V' := \{w \in V \mid w \preceq_T v\}$. Thus $T(v)$ has leaf set $L(T(v))$.

For a non-empty subset $L' \subseteq L$ of leaves, *the last common ancestor of* L', denoted as $\mathrm{lca}_T(L')$, is the unique $\preceq_T$-minimal vertex of T that is an ancestor of every vertex in L'. We will make use of the simplified notation $\mathrm{lca}_T(x_1, \ldots, x_k) := \mathrm{lca}_T(\{x_1, \ldots, x_k\})$ for a set $A = \{x_1, \ldots, x_k\}$ of vertices. The definition of $\mathrm{lca}_T(A)$ is conveniently extended to edges by setting $\mathrm{lca}_T(x, e) := \mathrm{lca}_T(\{x\} \cup e)$ and $\mathrm{lca}_T(e, f) := \mathrm{lca}_T(e \cup f)$, where the edges $e, f \in E(T)$ are simply treated as sets of vertices. We note for later reference that $\mathrm{lca}_T(A \cup B) = \mathrm{lca}_T(\mathrm{lca}_T(A), \mathrm{lca}_T(B))$ holds for non-empty vertex sets A, B of a tree. For simplicity the explicit reference to T is omitted whenever it is clear which tree is considered. Analogously, we often write $\mathsf{par}(v)$ and $\mathsf{child}(v)$ instead of $\mathsf{par}_T(v)$ and $\mathsf{child}_T(v)$ for $v \in V$.

3.3.3 *Edge Contraction, Restriction, and Refinement*

The *contraction* of an edge $e = uv$ in a tree $T = (V, E)$ refers to the removal of e and identification of u and v. We denote by T_e the tree that is obtained from T by contraction of e. Analogously, T_A is obtained by contracting a sequence of edges $A = (e_1, \ldots, e_k) \subseteq E$. We say that a rooted tree T on L *displays* a rooted tree T' on L', in symbols $T' \leq T$, if T' can be obtained from $T(\mathrm{lca}_T(L'))$ by a sequence of edge contractions. We write $T' < T$ if $T' \leq T$ and $T' \neq T$. The *restriction* $T_{|L'}$ of T to L' is the rooted tree obtained from $T(\mathrm{lca}_T(L'))$ by suppressing all vertices of degree 2 with the exception of the

root ρ_T if $\rho_T \in V(T(\mathrm{lca}_T(L')))$. By construction, $T_{|L'}$ is a phylogenetic tree. The suppression of vertices of degree 2 can be achieved by contraction of one of the adjacent edges. Moreover, $T_{|L'} \leq T$, i.e., T displays the restrictions $T_{|L'}$ to all subsets $L' \subseteq L$. Note that $T_{|L} = T$ if and only if T is phylogenetic; otherwise $T_{|L} < T$.

Moreover, we define $\mathcal{C}(T) := \{L(T(v)) \mid v \in V(T)\}$. A rooted tree is phylogenetic if and only if $L(T(u)) = L(T(v))$ implies $u = v$ for all $u, v \in V(T)$. We say that a rooted tree T' on L *refines* a rooted tree T on L if T' displays T. In particular, a phylogenetic tree T' on L refines a rooted tree T if and only if $\mathcal{C}(T) \subseteq \mathcal{C}(T')$.

For a leaf-colored tree T on L with coloring map $\sigma : L \to S$, in symbols (T, σ), we say that (T, σ) *displays* or is a *refinement* of (T', σ') if $T' \leq T$ and $\sigma'(v) = \sigma(v)$ for any $v \in L(T') \subseteq L$.

3.3.4 *Hierarchies*

A set system $\mathcal{C} \subseteq 2^L$ is called a *hierarchy* on a finite set L if

(i) either $A \subseteq B$, $B \subseteq A$, or $A \cap B = \emptyset$ for all $A, B \in \mathcal{C}$, and

(ii) $L \in \mathcal{C}$.

There exists a well-known one-to-one correspondence between rooted trees on L and hierarchies on L [202]:

Theorem 3.1. *Let $\mathcal{C}$ be a collection of non-empty subsets of a finite set L. Then there is a rooted tree T with $L(T) = L$ such that $\mathcal{C} = \mathcal{C}(T)$ if and only if $\mathcal{C}$ is a hierarchy on L. Moreover, if such a tree exists, it is, up to isomorphism, unique.*

3.3.5 *Triples, Consistency, and the Closure Operation*

Rooted triples are binary rooted phylogenetic trees on three leaves. We write $ab|c$ for the rooted triple with leaves a, b, and c if the path from its root to c does not intersect the path from a to b. The definition of "display" implies that a triple $ab|c$ with $a, b, c \in L$ is *displayed* by a rooted tree T if $\mathrm{lca}(a, b) \prec_T \mathrm{lca}(a, b, c)$.

The set of all triples that are displayed by T is denoted by $r(T)$. For a set R of rooted triples we define $R_x \subseteq R$ as the set of triples in R that contain the leaf x. A set of rooted triples R is called *consistent* if there exists a phylogenetic tree T on $L_R := \bigcup_{ab|c \in R} \{a, b, c\}$ that displays R, i.e., $R \subseteq r(T)$. In particular, a tree can display at most one triple on any set of three leaves. Thus a triple set R is inconsistent whenever $ab|c, ac|b \in R$. However, triple sets can be inconsistent even if they do not contain two triples on the same three leaves. Analogously, we say that a set of trees $\mathcal{T}$ is consistent it there is a tree T such that T displays every tree $T' \in \mathcal{T}$. Consistency of a set of triples R and more generally trees $\mathcal{T}$ can be decided in polynomial time by explicitly constructing a supertree T that displays all trees in $\mathcal{T}$ (see Section 3.4).

The requirement that a set R of triples is consistent, and thus, that there is a tree displaying all triples, makes it possible to infer new triples from the trees that display R and to define a *closure operation* for R [84, 26, 200, 25]. Let $co(R)$ be the set of all rooted trees with leaf set L_R that display R. The closure of a consistent set of rooted triples R is defined as

$$\mathrm{cl}(R) = \bigcap_{T \in co(R)} r(T).$$

Hence, a triple r is contained in the closure $\mathrm{cl}(R)$ if all trees that display R also display r. This operation satisfies the usual three properties of a closure operator [26], namely: (i) $R \subseteq \mathrm{cl}(R)$ (expansiveness), (ii) $R' \subseteq R$ implies that $\mathrm{cl}(R') \subseteq \mathrm{cl}(R)$ (isotony), and (iii) $\mathrm{cl}(\mathrm{cl}(R)) = \mathrm{cl}(R)$ (idempotency). Since $T \in co(r(T))$, it is easy to see that $\mathrm{cl}(r(T)) = r(T)$ and thus, $r(T)$ is always closed.

A set of rooted triples R *identifies* a tree T with leaf set L_R if R is displayed by T and every other tree T' that displays R is a refinement of T. A rooted triple $ab|c \in r(T)$ *distinguishes* an edge uv in T if and only if a, b, and c are descendants of u, v is an ancestor of a and b but not of c, and there is no descendant v' of v for which a and b are both descendants. In other words, $ab|c \in r(T)$ distinguishes the edge uv if and only if $\mathrm{lca}(a, b) = v$ and $\mathrm{lca}(a, b, c) = u$.

3.4 Aho graphs, Aho trees and the BUILD algorithm

Rooted triples are widely used in the context of supertree reconstruction because every phylogenetic tree T is identified by its triple set $r(T)$, and $r(T) \subseteq r(T')$ if and only if T' displays T [202]. As a consequence, supertree reconstruction can be phrased in terms of triples. As shown in [3] there is a polynomial time algorithm, usually referred to as BUILD [202, 212], that takes a set R of triples as input and either returns a particular phylogenetic tree $\mathrm{Aho}(R)$ that displays R, or recognizes R as inconsistent.

BUILD makes use of a simple graph representation of certain subsets of triples: Given a triple set R and a subset of leaves $L' \subseteq L$, the *Aho graph* $[R, L']$ has vertex set L' and there is an edge between two vertices $x, y \in L'$ if and only if there exists a triple $xy|z \in R$ with $z \in L'$ [3]. It is well known that R is consistent if and only if $[R, L']$ is disconnected for every subset $L' \subseteq L$ with $|L'| > 1$ [26]. BUILD uses Aho graphs in a top-down recursion: First, $[R, L]$ is computed and a tree T consisting only of the root ρ_T is initialized. If $[R, L]$ is connected and $|L| > 1$, then BUILD terminates and returns *"R is not consistent"*. Otherwise, BUILD adds the connected components $C_1, \ldots, C_k$ of $[R, L]$ as vertices to T and inserts the edges (ρ_T, C_i), $1 \leq i \leq k$. BUILD recurses on the Aho graphs $[R, C_i]$ (where vertex C_i in T plays the role of ρ_T) until it arrives at single-vertex components. This workflow is illustrated in Fig. 3. BUILD either returns the tree T or identifies the triple set R as "not consistent". Since the Aho graphs $[R, L']$ and their connected components are uniquely defined in each step of BUILD, the tree T is uniquely defined by R whenever it exists. T is known as the *Aho tree* and will be denoted by $\mathrm{Aho}(R)$.

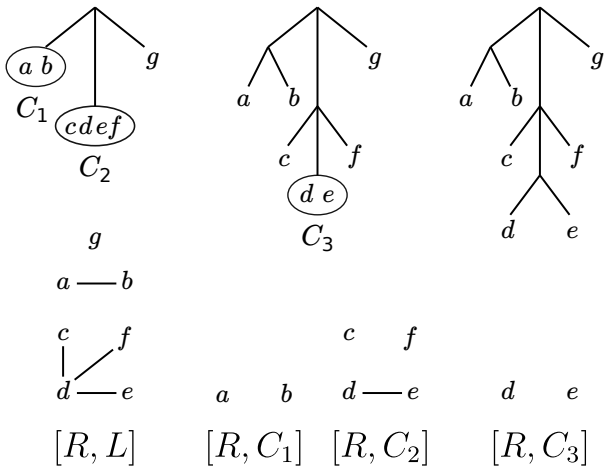


Fig. 3. Illustration of the `BUILD` Algorithm for a given set of leaves $L = \{a, b, c, d, e, f, g\}$ and triple set $R = \{ab|c, cd|g, de|c, df|a\}$. Shown is the Aho graph (bottom) together with the corresponding tree T (top) after each step of the recursion. The connected components C_i are represented as a "bag" of the vertices which they contain. The tree on the right is the final Aho tree.

This tree displays R and is least resolved in the sense that none of the edges in $\mathrm{Aho}(R)$ can be contracted without loosing a triple from R.
We will make use of the following result from [84] that is closely related to the `BUILD` Algorithm.

Lemma 3.1. *Let T be a phylogenetic tree and let R be a set of rooted triples. Then, R identifies T if and only if* $\mathrm{cl}(R) = r(T)$. *Moreover, if R identifies T, then* $\mathrm{Aho}(R) = T$.

3.5 COGRAPHS

Cographs form a class of undirected graphs that play an important in the context of this contribution. They are defined recursively [38]:

Definition 3.1. *An undirected graph G is a cograph if*

(1) $G = K_1$,

(2) $G = H \triangledown H'$, *where H and H' are cographs, or*

(3) $G = H \uplus H'$, *where H and H' are cograph.*

A graph is a cograph if and only if it does not contain an induced P_4, i.e., an induced path on four vertices [38].

Each cograph G is associated with *cotrees* T, that is, phylogenetic trees with inner vertices labeled by 0 or 1, whose leaves correspond to the vertices of G. In T, each subtree rooted at an inner vertex u with label 0 corresponds to the disjoint union of the subgraphs of G induced by the leaf sets $L(T(v))$ of the children $v \in \mathsf{child}(u)$ of u, and each subtree rooted at an inner vertex u with label 1 corresponds to the join of the subgraphs of G induced by the sets $L(T(v))$, $v \in \mathsf{child}(u)$. In other words (T, t) is a cotree for G if $t(\mathrm{lca}_T(x, y)) = 1$ if and only if $xy \in E(G)$. For each cograph G there is a unique *discriminating* cotree T with the property that the labels 0 and 1 alternate along each root-leaf

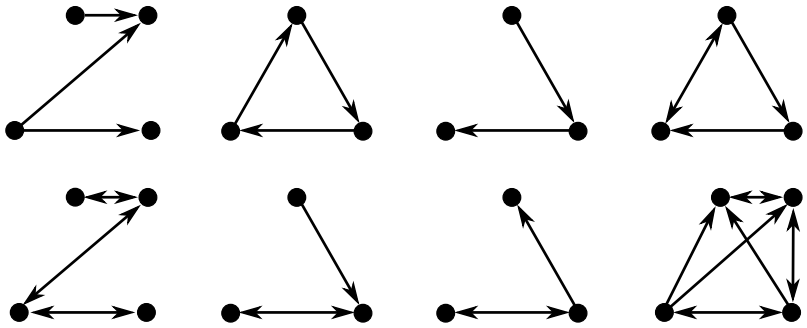


Fig. 4. The set of forbidden induced subgraphs that characterize di-cographs.

path in T [38]. For later reference, we summarize here some of the results from Hellmuth et al. [97, Section 3]:

Proposition 3.1. *Any cotree of a cograph G is a refinement of the unique discriminating cotree of G. In particular, (T_e, t_e) is a cotree for a cograph G if and only if (T, t) is a cotree for G, where $e = xy \in E(T)$ is an edge with $t(x) = t(y)$ that is contracted to the vertex v_e in T_e with $t_e(v_e) = t(x)$ and $t_e(v) = t(v)$ for all remaining inner vertices $v \neq v_e$.*

The concept of cographs can be generalized to directed graphs by adding order compositions. More precisely, a directed graph is a *di-cograph* if it is either the K_1 or it is obtained from two or more di-cographs by a disjoint union, a join, or an order composition. Di-cographs are characterized by the eight forbidden subgraphs shown in Fig. 4 [60, 40].
Similarly to cographs, every di-cograph $\vec{G}$ is explained by a unique discriminating cotree $(T, \vec{t})$ [162, 157], that is, an ordered phylogenetic tree T with leaf set $V(\vec{G})$ and a vertex labeling function $\vec{t}\colon V^0(T) \to \{0, 1, \vec{1}\}$, such that $\vec{t}(u) \neq \vec{t}(v)$ for all inner edges uv in T, defined by

$$\vec{t}(\mathrm{lca}(x,y)) = \begin{cases} 0, & \text{if } (x,y),(y,x) \notin E(\vec{G}) \\ 1, & \text{if } (x,y),(y,x) \in E(\vec{G}) \\ \vec{1}, & \text{else .} \end{cases}$$

Since the vertices in the cotree T are ordered, the label $\vec{1}$ on some $\mathrm{lca}(x, y)$ of two distinct leaves $x, y \in L$ means that there is an edge $(x, y) \in E(\vec{G})$, while $(y, x) \notin E(\vec{G})$, whenever x is placed to the left of y in T [100].

4 BEST MATCH GRAPHS

Reciprocal best hits (RBH) are the most commonly employed method for inferring orthologs [6, 10]. Practical applications typically produce, for each gene from species A, a list of genes found in species B, ranked in the order of decreasing sequence similarity. From these lists, reciprocal best hits are readily obtained. Some software tools, such as `ProteinOrtho` [144, 145], explicitly construct a digraph whose arcs are the (approximately) co-optimal best matches. Empirically, the pairs of genes that are identified as reciprocal best hits depend on the details of the computational method for quantifying sequence similarity (see Section 2.5 for more details). Independent of the computational details, however, reciprocal best matches are of interest because they approximate the concept of pairs of *reciprocal evolutionarily most closely related* genes. It is this notion that links best matches directly to orthology: Given a gene x in species a (and disregarding horizontal gene transfer), all its co-orthologous genes y in species b are by definition closest relatives of x.

The purpose of this chapter is to establish a characterization of BMGs as an indispensable prerequisite for any method that attempts to directly correct empirical best match data. We start by formally introducing the best match relation in Section 4.1 before establishing in Section 4.2 a few simple properties of BMGs and show that key problems can be broken down to the connected components of 2-colored BMGs. These are considered in detail in Section 4.3. The characterization of 2-BMGs is not a trivial task. Although the existence of at least one out-neighbor for each vertex is an obvious necessary condition, the example in Fig. 6 shows that it is not sufficient. In Section 4.3 we prove our main results on 2-BMGs: the existence of a unique least resolved tree that explains any given 2-BMG (Thm. 4.2), a characterization in terms of informative triples that can be extracted directly from the input graph (Thm. 4.6), and a characterization in terms of three simple conditions on the out-neighborhoods (Thm. 4.4). Section 4.4 provides a complete characterization of a general BMG: It is necessary and sufficient that the subgraph induced by each pair of colors is a 2-BMG and that the union of the triple sets of their least resolved tree representations is consistent. This chapter is finally closed with a brief discussion of algorithmic considerations in Section 4.5.
The results presented here have been previously published in Geiß et al. [73].

4.1 INTRODUCTION OF THE BEST MATCH RELATION

Evolutionary relatedness is a phylogenetic property and thus is defined relative to the phylogenetic tree T of the genes under consideration. More precisely, we consider a set of genes L (the leaves of the phylogenetic tree T), a set of species S, and a map σ assigning to each gene $x \in L$ the species $\sigma(x) \in S$ within which

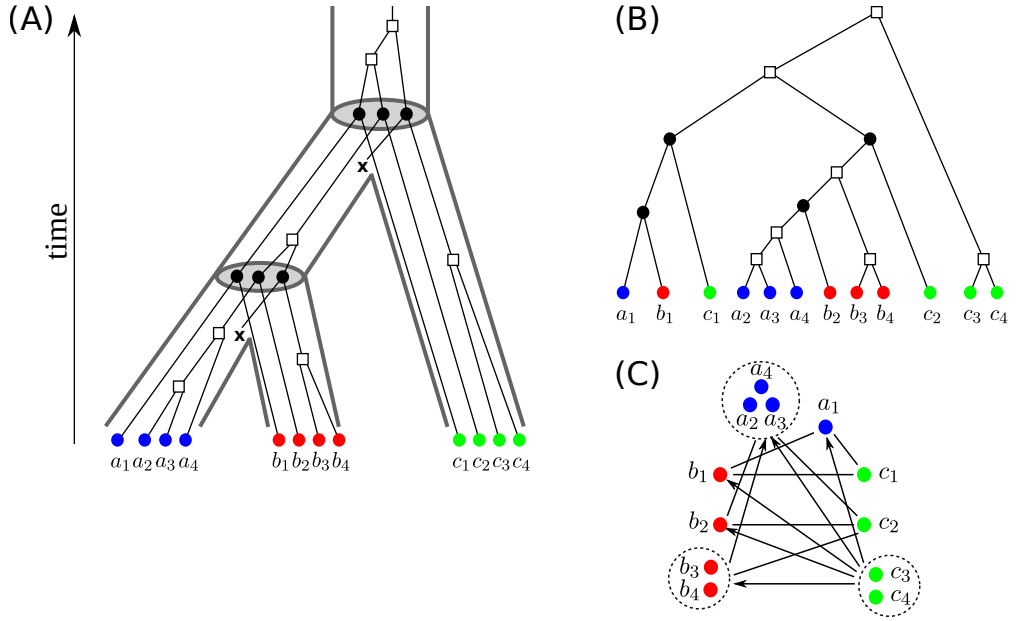


Fig. 5. (A) An evolutionary scenario consisting of a gene tree whose inner vertices are marked by the event type (● for speciations, □ for gene duplications, and × for gene loss) together with its embedding into a species tree (drawn as tube-like outline). All events are placed on a time axis. (B) The observable part of the gene tree (T, σ) obtained from the gene tree in the full evolutionary scenario by removing all leaves marked as loss events and suppression of all resulting vertices of degree 2 [105, 99]. (C) The colored best match graph $(\vec{G}, \sigma)$ that is explained by (T, σ). Directed arcs indicate the best match relation $\rightarrow$. Bi-directional best matches ($x \rightarrow y$ and $y \rightarrow x$) are drawn as solid lines without arrow heads instead of pairs of arrows. Dotted circles collect sets of leaves that have the same in- and out-neighborhood. The corresponding arcs are shown only once.

it resides. A gene x is more closely related to some gene y than to another gene z if $\mathrm{lca}(x, y) \prec \mathrm{lca}(x, z)$. We can now make the notion of a *best match* precise:

Definition 4.1. *Consider a tree T with leaf set L and a surjective map $\sigma : L \to S$. Then $y \in L$ is a* best match *of $x \in L$, in symbols $x \to y$, if and only if $\mathrm{lca}(x, y) \preceq \mathrm{lca}(x, y')$ holds for all leaves y' from species $\sigma(y') = \sigma(y)$.*

In order to understand how best matches (in the sense of Def. 4.1) are approximated by best hits computed by mean sequence similarity, we first observe that best matches can be expressed in terms of the evolutionary time. More precisely, the evolutionary relatedness of two taxa x and y is most directly expressed by the divergence time $\tau(x, y)$, which is the total time elapsed in both lineages since the last common ancestor of x and y in the corresponding gene tree T, as in Fig. 5. Here, we consider only the case that all leaves refer to extant genes or taxa, i.e., $\tau(x, y) = 2\hat{\tau}(\mathrm{lca}(x, y))$, where $\hat{\tau}$ is the age of $\mathrm{lca}(x, y)$. The best match relation $\rightarrow$ can thus also be defined in terms of divergence time: $x \rightarrow y$ if and only if

$$y \in \arg\min_{y' \in L[t]} \tau(x, y'). \tag{1}$$

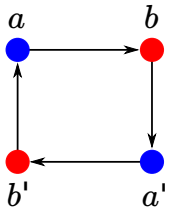


Fig. 6. Not every graph with non-empty out-neighborhoods is a colored best match graph. The 4-vertex graph $(\vec{G}, \sigma)$ with two colors shown here is the smallest connected counterexample: there is no leaf-colored tree (T, σ) that explains $(\vec{G}, \sigma)$.

Mathematically, this is equivalent to Def. 4.1 whenever τ is an ultrametric distance on T. For the divergence time τ this is by definition the case. Best match heuristics therefore assume (often tacitly) that the *molecular clock hypothesis* [241, 137] is at least a reasonable approximation. In Section 2.5 it was already discussed that, although the assumption of a molecular clock is often violated, best match heuristics still perform equally good or even better than phylogenetic methods but there also a need for identifying incorrect best match assignments.

Extending the information used for the correction of initial reciprocal best hits to a global scale, it is possible to improve orthology prediction by enforcing the global cograph of the orthology relation [98, 141]. The following three chapters originated from an analogous question: Can empirical (reciprocal) best match data be improved just by using the fact that ideally a (reciprocal) best match relation should derive from a tree T according to Def. 4.1? To answer this question we first need to understand the structure of best match relations.
The best match relation is conveniently represented as a colored digraph:

Definition 4.2. *Given a tree T with leaf set L and a map $\sigma : L \to S$, the* colored best match graph *(BMG) $\vec{G}(T, \sigma)$ has vertex set L and arcs $(x, y) \in E(\vec{G})$ if $x \neq y$ and $x \to y$. Each vertex $x \in L$ obtains the color $\sigma(x)$.*
The rooted tree T explains *the vertex-colored graph $(\vec{G}, \sigma)$ if $(\vec{G}, \sigma)$ is isomorphic to the BMG $\vec{G}(T, \sigma)$.*

To emphasize the number of colors used in $\vec{G}(T, \sigma)$, that is, the number of species in S, we will write $|S|$-BMG. Note in particular that, for $|S| = 1$, the edge-less graphs are explained by any tree. Hence, we will assume $|S| \geq 2$ in the following to avoid dealing with trivial cases.
In particular, Def. 4.2 immediately implies

Observation 4.1. *If $(\vec{G}, \sigma)$ is a BMG, then σ is a proper vertex coloring.*

Hence, a colored digraph $(\vec{G}, \sigma)$ can only be explained by a leaf-colored tree if σ is a proper vertex coloring. We may thus assume throughout this chapter that $(\vec{G}, \sigma)$ is a properly vertex-colored graph.

4.2 Basic properties of best match relations

The best match relation $\to$ is reflexive because $\mathrm{lca}(x, x) = x \prec \mathrm{lca}(x, y)$ for all genes y with $\sigma(x) = \sigma(y)$. For any pair of distinct genes x and y with $\sigma(x) = \sigma(y)$ we have $\mathrm{lca}(x, y) \notin \{x, y\}$, hence the relation $\to$ has off-diagonal

pairs only between genes from different species. There is still a one-to-one correspondence between BMGs (Def. 4.2) and best match relations (Def. 4.1): In the BMG the reflexive loops are omitted, in the relation $\to$ they are added.

Note that the tree $(\vec{G}, \sigma)$ and the corresponding BMG $\vec{G}(T, \sigma)$ employ the same coloring map $\sigma : L \to S$. Recall in this context that we do not distinguish between isomorphic vertex-colored graphs.

4.2.1 *Thinness*

In undirected graphs, equivalence classes of vertices that share the same neighborhood are considered in the context of thinness of the graph [159, 215, 27]. The concept naturally extends to digraphs [93]. For our purposes the following variation on the theme is most useful:

Definition 4.3. *Given a digraph $\vec{G}$, two vertices $x, y \in V(\vec{G})$ are in relation* R *if $N^+(x) = N^+(y)$ and $N^-(x) = N^-(y)$.*

For each R-class α we have $N^+(x) = N^+(\alpha)$ and $N^-(x) = N^-(\alpha)$ for all $x \in \alpha$. It is obvious, therefore, that R is an equivalence relation on the vertex set of $\vec{G}$. Moreover, since we consider loop-free graphs, one can easily see that $\vec{G}[\alpha]$ is always edge-less. We write $\mathcal{N}$, or $\mathcal{N}(\vec{G})$, for the corresponding partition, i.e., the set of R-classes of $\vec{G}$. Individual R-classes will be denoted by lowercase Greek letters. Note that for the graphs considered here, we always have $N^+_{\sigma(x)}(x) = N^-_{\sigma(x)}(x) = \emptyset$. When considering sets $N^+_s(x)$ and $N^-_s(x)$, we can therefore always assume $s \neq \sigma(x)$. Furthermore, $\mathcal{N}_s$ denotes the set of R-classes with color s.
By construction, the function $N^+ : V(\vec{G}) \to 2^{V(\vec{G})}$ is isotonic, i.e., $A \subseteq B$ implies $N^+(A) \subseteq N^+(B)$. In particular, therefore, we have for $\alpha, \beta \in \mathcal{N}$:

(i) $\alpha \subseteq N^+(\beta)$ implies $N^+(\alpha) \subseteq N^+(N^+(\beta))$

(ii) $N^+(\alpha) \subseteq N^+(\beta)$ implies $N^+(N^+(\alpha)) \subseteq N^+(N^+(\beta))$.

These observations will be useful in the proofs below.
By construction, every vertex in a BMG has at least one out-neighbor of every color except its own, i.e., $|N^+(x)| \geq |S| - 1$ holds for all x. In contrast, $N^-(x) = \emptyset$ is possible.

4.2.2 *Some Simple Observations*

The color classes $L[s]$ on the leaves L of a leaf-labeled tree (T, σ) are independent sets in $\vec{G}(T, \sigma)$ since arcs in $\vec{G}(T, \sigma)$ connect only vertices with different colors. For any pair of colors $s, t \in S$, therefore, the induced subgraph $\vec{G}[L[s] \cup L[t]]$ of $\vec{G}(T, \sigma)$ is bipartite. Since the definition of $x \to y$ does not depend on the presence or absence of vertices u with $\sigma(u) \notin \{\sigma(x), \sigma(y)\}$, we have

Observation 4.2. *Let $(\vec{G}, \sigma)$ be a BMG explained by (T, σ) and let $L' := \bigcup_{s \in S'} L[s]$ be the subset of vertices with a restricted color set $S' \subseteq S$. Then the*

induced subgraph $(\vec{G}[L'], \sigma_{|L'})$ *is explained by the restriction* $T_{L'}$ *of* T *to the leaf set* L'.

It follows in particular that $\vec{G}_{st} = \vec{G}[L[s] \cup L[t]]$ is explained by the restriction $T_{st} = T_{|L[s]\cup L[t]}$ of T to the colors s and t. Furthermore, $\vec{G}$ is the edge-disjoint union of bipartite subgraphs corresponding to color pairs, i.e.,

$$E(\vec{G}) = \bigcup_{\{s,t\}\in\binom{S}{2}} E(\vec{G}_{st}).$$

In order to understand when arbitrary graphs $(\vec{G}, \sigma)$ are BMGs, it is sufficient, therefore, to characterize 2-BMGs. A formal proof will be given later on in Section 4.4.

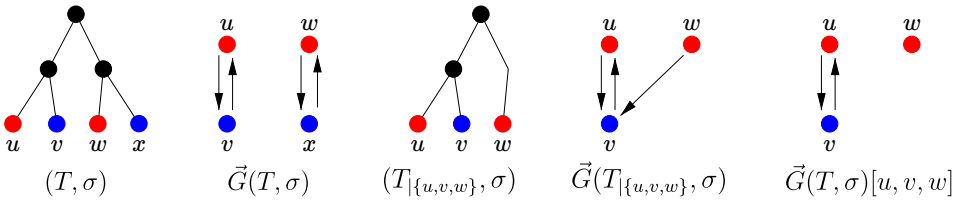


Fig. 7. $T_{|\{u,v,w\}}$ is displayed by T but $\vec{G}(T_{|\{u,v,w\}}, \sigma)$ is not isomorphic to the induced subgraph $\vec{G}(T, \sigma)[u, v, w]$ of $\vec{G}(T, \sigma)$, since $\vec{G}(T_{|\{u,v,w\}}, \sigma)$ contains the additional arc (w, v).

Note, the condition that "T explains $(\vec{G}, \sigma)$" does not imply that $(T_{L'}, \sigma_{|L'})$ explains $(\vec{G}[L'], \sigma_{|L'})$ for arbitrary subsets of $L' \subseteq L$. Fig. 7 shows that, indeed, not every induced subgraph of a BMG is necessarily a BMG. However, we have the following, weaker property:

Lemma 4.1. *Let* $(\vec{G}, \sigma)$ *be a BMG with vertex set* L *explained by* (T, σ) *and let* $(T_{|L'}, \sigma_{|L'})$ *be the restriction of* (T, σ) *to* $L' \subseteq L$. *Then the induced subgraph* $(\vec{G}[L'], \sigma_{|L'})$ *of* $(\vec{G}, \sigma)$ *is a (not necessarily induced) subgraph of* $\vec{G}(T_{|L'}, \sigma_{|L'})$.

Proof. If $(u, v) \in E(\vec{G})$ and $u, v \in L'$, then $\mathrm{lca}_T(u, v) \preceq_T \mathrm{lca}_T(u, z)$ for all $z \in L[\sigma(v)]$ and thus, the inequality $\mathrm{lca}_{T'}(u, v) \preceq_{T'} \mathrm{lca}_{T'}(u, z)$ is in particular true for all $z \in L' \cap L[\sigma(v)] = L'[\sigma(v)]$. Hence, $u, v \in L'$ and $(u, v) \in E(\vec{G})$ implies $(u, v) \in E(\vec{G}[L'])$, which concludes the proof. □

4.2.3 *Connectedness*

We briefly present some results concerning the connectedness of BMGs. In particular, it turns out that connected BMGs have a simple characterization in terms of their representing trees.

Theorem 4.1. *Let* (T, σ) *be a leaf-labeled 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 *of* T *such that* $\sigma(L(T(v))) \neq S$. *Furthermore, if* $\vec{G}(T, \sigma)$ *is not connected, then for every connected component* C *of* $\vec{G}(T, \sigma)$ *there is a child* v *of the root* ρ *such that* $V(C) \subseteq L(T(v))$.

Proof. For convenience we write $L_v := L(T(v))$. Suppose $\sigma(L_v) = S$ holds for all children v of the root. Then for any pair of colors $s, t \in S$ we find for a leaf $x \in L_v$ with $\sigma(x) = s$ a leaf $y \in L_v$ with $\sigma(y) = t$ within $T(v)$; thus $\operatorname{lca}(x, y)$ is in $T(v)$ and therefore $\operatorname{lca}(x, y) \prec \rho$. Hence, all best matching pairs are confined to the subtrees below the children of the root. The corresponding leaf sets are thus mutually disconnected in $\vec{G}(T, \sigma)$.

Conversely, suppose that one of the children v of the root ρ satisfies $\sigma(L_v) \neq S$. Therefore there is a color $t \in S$ with $t \notin \sigma(L_v)$. Then for every $x \in L_v$ there is an arc $x \to z$ for all $z \in L[t]$ since for all such z we have $\operatorname{lca}(x, z) = \rho$. If $L[t] = L \setminus L_v$, we can conclude that $\vec{G}(T, \sigma)$ is a connected digraph. Otherwise, every leaf $y \in L \setminus L_v$ with color $\sigma(y) \neq t$ has an out-arc $y \to z$ to some $z \in L[t]$ and thus, there is a path $y \to z \leftarrow x$ connecting $y \in L \setminus L_v$ to every $x \in L_v$. Finally, for any two vertices $y, y' \in L \setminus (L_v \cup L[t])$ there are vertices $z, z' \in L[t]$ such that arcs exist that form a path $y \to z \leftarrow x \to z' \leftarrow y'$ connecting z with z' and both to any $x \in L_v$. In summary, therefore, $\vec{G}(T, \sigma)$ is a connected digraph.

For the last statement, we argue as above and conclude that if $\sigma(L_v) = S$ for all children v of the root (or, equivalently, if $\vec{G}(T, \sigma)$ is not connected), then all best matching pairs are confined to the subtrees below the children of the root ρ. Thus the vertices of every connected component of $\vec{G}(T, \sigma)$ must be leaves of a subtree $T(v)$ for some child v of the root ρ. □

The following result shows that BMGs can be characterized by their connected components: the disjoint union of vertex disjoint BMGs is again a BMG if and only if they all share the same color set. It suffices, therefore, to consider each connected component separately.

Proposition 4.1. *Let $(\vec{G}_i, \sigma_i)$ be vertex disjoint BMGs with vertex sets L_i and color sets $S_i = \sigma_i(L_i)$ for $1 \leq i \leq k$. Then the disjoint union $(\vec{G}, \sigma) := \bigcup_{i=1}^{k} (\vec{G}_i, \sigma_i)$ is a BMG if and only if all color sets are the same, i.e., $\sigma_i(L_i) = \sigma_j(L_j)$ for $1 \leq i, j \leq k$.*

Proof. The statement is trivially fulfilled for $k = 1$. For $k \geq 2$, the disjoint union $(\vec{G}, \sigma)$ is not connected. Assume that $\sigma_i(L_i) = \sigma_j(L_j)$ for all i, j. Let (T_i, σ_i) be trees explaining $(\vec{G}_i, \sigma_i)$ for $1 \leq i \leq k$. We construct a tree (T, σ) as follows: Let ρ be the root of (T, σ) with children $r_1, \ldots r_k$. Then we identify r_i with the root of T_i and retain all leaf colors. In order to show that (T, σ) explains $(\vec{G}, \sigma)$, we recall from Thm. 4.1 that all best matching pairs are confined to the subtrees below the children of the root and hence, each connected component of $(\vec{G}, \sigma)$ forms a subset of one of the leaf sets L_i. Since each (T_i, σ_i) explains $(\vec{G}_i, \sigma_i)$, we conclude that the BMG explained by (T, σ) is indeed the disjoint union of the $(\vec{G}_i, \sigma_i)$, i.e., $(\vec{G}, \sigma)$. Thus $(\vec{G}, \sigma)$ is a BMG.

Conversely, assume that $(\vec{G}, \sigma)$ is a BMG but $\sigma_i(L_i) \neq \sigma_k(L_k)$ for some $k \neq i$. By construction, $\sigma(L_i) = \sigma_i(L_i)$ and $\sigma(L_k) = \sigma_k(L_k)$. In particular, for every color $t \notin \sigma(L_i)$ and every vertex $x \in L_i$, there is a $j \neq i$ with $t \in \sigma(L_j)$ such that there exists an outgoing arc form x to some vertex $y \in L_j$ with color $\sigma(y) = t$. Thus (x, y) is an arc connecting L_i with some L_j, $j \neq i$, contradicting the assumption that each L_i forms a connected component of $(\vec{G}, \sigma)$. Hence, the color sets cannot differ between connected components. □

The example $\vec{G}(T_{\{u,v,w\}}, \sigma)$ in Fig. 7 already shows, however, that $\vec{G}(T, \sigma)$ is not necessarily strongly connected.

4.3 TWO-COLORED BEST MATCH GRAPHS (2-BMGS)

As we have already argued in the previous section, understanding 2-BMGs is crucial for the characterization of BMGs with an arbitrary number of colors. In this section we derive two characterizations for 2-BMGs: one in terms of informative triples that can be inferred directly from the directed graph in question, and one in terms of out-neighborhoods. Moreover, we show that for any given 2-BMG there exists a unique least resolved tree explaining it. We start with some basic properties of R-classes in 2-BMGs and so-called *roots of* R*-classes*.

Throughout this section we assume that $\sigma(L) = \{s, t\}$ ***contains exactly two colors.***

4.3.1 *Thinness Classes*

A connected 2-BMG $(\vec{G}, \sigma)$ contains at least two R-classes since all in- and out-neighbors y of $x \in V(\vec{G})$, by construction, have a color $\sigma(y)$ different from $\sigma(x)$. Consequently, any 2-BMG is bipartite. Furthermore, if $\sigma(x) \neq \sigma(y)$, then $N^+(x) \cap N^+(y) = \emptyset$. Since $N^+(x) \neq \emptyset$ and all members of $N^+(x)$ have the same color, we observe that $N^+(x) = N^+(y)$ implies $\sigma(x) = \sigma(y)$. By a slight abuse of notation we will often write $\sigma(x) = \sigma(\alpha)$ for an element x of some R-class α. Two leaves x and y of the same color that have the same last common ancestor with all other leaves in T, i.e., that satisfy $\mathrm{lca}(x, u) = \mathrm{lca}(y, u)$ for all $u \in L \setminus \{x, y\}$, by construction, have the same in-neighbors and the same out-neighbors in $\vec{G}(T, \sigma)$, hence $x\mathsf{R}y$.

Observation 4.3. *Let* $(\vec{G}, \sigma)$ *be a connected 2-BMG and* $\alpha \in \mathcal{N}$ *an* R*-class. Then* $\sigma(x) = \sigma(y)$ *for any* $x, y \in \alpha$.

The following result shows that the out-neighborhood of any R-class is a disjoint union of R-classes.

Lemma 4.2. *Let* $(\vec{G}, \sigma)$ *be a connected 2-BMG. Then any two* R*-classes* $\alpha, \beta \in \mathcal{N}$ *satisfy*

(N0) $\beta \subseteq N^+(\alpha)$ *or* $\beta \cap N^+(\alpha) = \emptyset$.

Proof. For any $y \in \beta$, the definition of R-classes implies that $y \in N^+(\alpha)$ if and only if $\beta \subseteq N^+(\alpha)$. Hence, either all or none of the elements of β are contained in $N^+(\alpha)$. □

The connection between the R-classes of $\vec{G}(T, \sigma)$ and the tree (T, σ) is captured by identifying an internal node in T that is, as we shall see, in a certain sense characteristic for a given equivalence class:

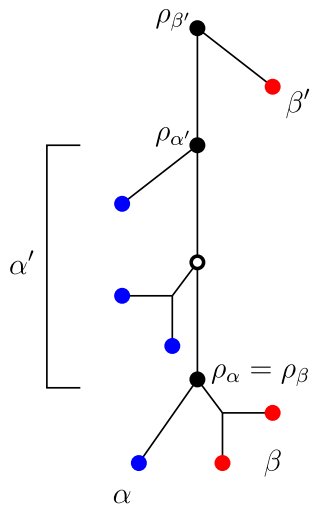


Fig. 8. Relationship between R-classes and their roots. A tree with two colors (blue and red) and four R-classes α, α' (blue) and β, β' (red) together with their corresponding roots ρ_α, $\rho_{\alpha'}$, ρ_β, and $\rho_{\beta'}$ are shown.

Definition 4.4. *For a leaf-labeled tree (T, σ) and the corresponding BMG $\vec{G}(T, \sigma)$, the* root ρ_α of the R-class α *is given as*

$$\rho_\alpha = \max_{\substack{x \in \alpha \\ y \in N^+(\alpha)}} \operatorname{lca}(x, y).$$

For an illustration of this concept see Fig. 8.

Corollary 4.1. *Let (T, σ) be a leaf-labeled tree with corresponding BMG $\vec{G}(T, \sigma)$ and ρ_α the root of an R-class α. Then it holds for any $y \in N^+(\alpha)$:*

$$\rho_\alpha = \max_{x \in \alpha} \operatorname{lca}(x, y).$$

In particular, $\operatorname{lca}(x, y) = \operatorname{lca}(x, z)$ *for all* $y, z \in N^+(\alpha)$.

Proof. For any $y \in N^+(\alpha)$ it holds by definition of $N^+(\alpha)$ that $\operatorname{lca}(x, y) \preceq \operatorname{lca}(x, z)$ for $x \in \alpha$ and any z with $\sigma(z) = \sigma(y)$. This together with Observation 4.3 implies $\operatorname{lca}(x, y) = \operatorname{lca}(x, z)$ for any two $y, z \in N^+(\alpha)$ and $x \in \alpha$. □

The following lemma collects some simple properties of the roots of R-classes that will be useful for the proofs of the main results.

Lemma 4.3. *Let $(\vec{G}, \sigma)$ be a connected 2-BMG explained by (T, σ) and let α, β be R-classes with roots ρ_α and ρ_β, respectively. Then the following statements hold:*

(i) $\rho_\alpha \preceq \operatorname{lca}(\alpha, \beta)$ *and* $\rho_\beta \preceq \operatorname{lca}(\alpha, \beta)$*; equality holds for at least one of them if and only if ρ_α, ρ_β are comparable, i.e., $\rho_\alpha \preceq \rho_\beta$ or $\rho_\beta \preceq \rho_\alpha$.*

(ii) The subtree $T(\rho_\alpha)$ contains leaves of both colors.

(iii) $N^+(\alpha) \preceq \rho_\alpha$.

(iv) If $\beta \subseteq N^+(\alpha)$*, then* $\rho_\beta \preceq \rho_\alpha$.

(v) If $\rho_\alpha = \rho_\beta$ *and* $\alpha \neq \beta$*, then* $\sigma(\alpha) \neq \sigma(\beta)$.

(vi) $N^+(\alpha) = \{y \mid y \in L(T(\rho_\alpha)) \text{ and } \sigma(y) \neq \sigma(\alpha)\}$

(vii) $N^+(N^+(\alpha)) \preceq \rho_\alpha$.

Proof. (i) By Condition (N0) in Lemma 4.2 we have either $\beta \subseteq N^+(\alpha)$ or $\beta \cap N^+(\alpha) = \emptyset$. By definition of $N^+(\beta)$, we have $\mathrm{lca}(x', y) \preceq \mathrm{lca}(x, y)$, where $y \in \beta$, $x' \in N^+(\beta)$, and $x \in \alpha$. Therefore, if $\beta \subseteq N^+(\alpha)$, then $\rho_\beta = \max_{x' \in N^+(\beta)} \mathrm{lca}(x', \beta) \preceq \max_{x \in \alpha} \mathrm{lca}(x, \beta) = \mathrm{lca}(\alpha, \beta)$. Moreover, Cor. 4.1 implies $\rho_\alpha = \max_{y \in N^+(\alpha)} \mathrm{lca}(\alpha, y) = \max_{y \in \beta} \mathrm{lca}(\alpha, y) = \mathrm{lca}(\alpha, \beta)$.

If $\beta \cap N^+(\alpha) = \emptyset$, then $\mathrm{lca}(\alpha, y) \succ \max_{y' \in N^+(\alpha)} \mathrm{lca}(\alpha, y') = \rho_\alpha$ for all $y \in \beta$, i.e., $\mathrm{lca}(\alpha, \beta) \succ \rho_\alpha$. Moreover, by definition of ρ_β, we have $\rho_\beta = \max_{x \in N^+(\beta)} \mathrm{lca}(x, \beta) \preceq \max_{x \in \alpha} \mathrm{lca}(x, \beta) = \mathrm{lca}(\alpha, \beta)$.

Now assume that ρ_α and ρ_β are comparable. W.l.o.g. we assume $\rho_\alpha \succeq \rho_\beta$. Since $\alpha \preceq \rho_\alpha$ and $\beta \preceq \rho_\beta$ is true by definition, we obtain $\mathrm{lca}(\alpha, \beta) = \rho_\alpha \succeq \rho_\beta$. Conversely, if $\rho_\alpha = \mathrm{lca}(\alpha, \beta) \succeq \rho_\beta$, then ρ_α and ρ_β are necessarily comparable.

(ii) As argued above, $N^+(x) \neq \emptyset$ for all vertices x. Let $x \in \alpha$ and $y \in N^+(x)$ such that $\rho_\alpha = \mathrm{lca}(x, y)$. By definition, $\sigma(x) \neq \sigma(y)$. Since ρ_α is an ancestor of both x and y, the statement follows.

(iii) Since $T(\rho_\alpha)$ contains leaves of both colors, there is in particular a leaf y with $\sigma(y) \neq \sigma(x)$ within $T(\rho_\alpha)$. It satisfies $\mathrm{lca}(x, y) \preceq \rho_\alpha$ and thus all arcs going out from $x \in \alpha$ are confined to leaves of $T(\rho_\alpha)$, i.e., $N^+(\alpha) \preceq \rho_\alpha$.

(iv) is a direct consequence of (i) and (iii).

(v) Assume, for contradiction, that $\sigma(\alpha) = \sigma(\beta)$. As $N^+(\alpha) \neq \emptyset$, there is some $y \in N^+(\alpha)$ with $\mathrm{lca}(\alpha, y) = \rho_\alpha$. Since $\rho_\alpha = \rho_\beta = \mathrm{lca}(\alpha, \beta)$ by (i), we have $\mathrm{lca}(\alpha, y) \succeq \mathrm{lca}(\beta, y)$. By definition of ρ_β, there is a leaf $z \in N^+(\beta)$ such that $\mathrm{lca}(\beta, z) = \rho_\beta$. Thus $\mathrm{lca}(\beta, y) \preceq \mathrm{lca}(\beta, z)$, which implies that y is a best match of β, i.e., $y \in N^+(\beta)$. Hence, $N^+(\alpha) = N^+(\beta)$. On the other hand, as $\mathrm{lca}(\alpha, \beta) = \rho_\alpha$, we have $\mathrm{lca}(\alpha, y) = \mathrm{lca}(\beta, y)$ for any y with $\mathrm{lca}(\alpha, y) \succeq \rho_\alpha$. As a consequence, since $\rho_\alpha \preceq \mathrm{lca}(\alpha, y')$ for all $y' \in N^-(\alpha)$, it is true that $\mathrm{lca}(y', \beta) = \mathrm{lca}(y', \alpha) \preceq \mathrm{lca}(y', z)$ for all z with $\sigma(z) = \sigma(\alpha)$. Hence, $y \in N^-(\alpha)$ if and only if $y \in N^-(\beta)$. It follows $\alpha = \beta$; a contradiction.

(vi) Let $y \in N^+(\alpha)$. Then $\sigma(y) \neq \sigma(\alpha)$ by definition. In addition, we have $y \preceq \rho_\alpha$ by (iii). Conversely, suppose that $y \in L(T(\rho_\alpha))$ and $\sigma(y) \neq \sigma(\alpha)$. Since $y \in L(T(\rho_\alpha))$, it is true that $y, \alpha \preceq \rho_\alpha$ and therefore, $\mathrm{lca}(\alpha, y) \preceq \rho_\alpha$. By definition of the root of α, there exist $x' \in \alpha$ and $y' \in N^+(\alpha)$ such that $\rho_\alpha = \mathrm{lca}(x', y') \preceq \mathrm{lca}(x', z)$ for all z with $\sigma(z) = \sigma(y')$. Since $\mathrm{lca}(\alpha, y) \preceq \rho_\alpha$, this implies $y \in N^+(\alpha)$.

(vii) Lemma 4.2 and Property (iv) imply that $N^+(\alpha)$ is a disjoint union of R-classes γ with $\rho_\gamma \preceq \rho_\alpha$ and $\sigma(\gamma) \neq \sigma(\alpha)$. Thus $N^+(N^+(\alpha)) = \bigcup_{\substack{\gamma \in \mathcal{N} \\ \gamma \subseteq N^+(\alpha)}} N^+(\gamma) = N^+(\bigcup_{\substack{\gamma \in \mathcal{N} \\ \gamma \subseteq N^+(\alpha)}} \gamma)$. By (iii) and (iv), we have $N^+(\gamma) \preceq \rho_\alpha$ for any such γ, thus $N^+(N^+(\alpha)) \preceq \rho_\alpha$.

□

Property (N0) implies that there are four distinct ways in which two R-classes α and β with distinct colors can be related to each other. These cases distinguish the relative location of their roots ρ_α and ρ_β:

Lemma 4.4. *If $(\vec{G}, \sigma)$ is a connected 2-BMG explained by some tree (T, σ), and α, β are* R*-classes with $\sigma(\alpha) \neq \sigma(\beta)$, then exactly one of the following four cases is true:*

(i) $\alpha \subseteq N^+(\beta)$ *and* $\beta \subseteq N^+(\alpha)$*. In this case* $\rho_\alpha = \rho_\beta$*.*

(ii) $\alpha \subseteq N^+(\beta)$ *and* $\beta \cap N^+(\alpha) = \emptyset$*. In this case* $\rho_\alpha \prec \rho_\beta$*.*

(iii) $\beta \subseteq N^+(\alpha)$ *and* $\alpha \cap N^+(\beta) = \emptyset$*. In this case* $\rho_\beta \prec \rho_\alpha$*.*

(iv) $\alpha \cap N^+(\beta) = \beta \cap N^+(\alpha) = \emptyset$*. In this case* ρ_α *and* ρ_β *are not* $\preceq$*-comparable.*

Proof. Set $\sigma(\alpha) = s$ and $\sigma(\beta) = t$, $s \neq t$, and consider the roots ρ_α and ρ_β of the two R-classes. Then there are exactly four cases:

(i) For $\rho_\alpha = \rho_\beta$, Lemma 4.3(i) implies $\rho_\alpha = \rho_\beta = \mathrm{lca}(\alpha, \beta)$. By definition of ρ_α, $y \in N^+(\alpha)$ for all $y \in L(T(\rho_\alpha))$ with $\sigma(y) \neq \sigma(\alpha)$ by Lemma 4.3(vi). A similar result holds for ρ_β. It immediately follows $\alpha \subseteq N^+(\beta)$ and $\beta \subseteq N^+(\alpha)$.

(ii) In the case $\rho_\alpha \succ \rho_\beta$, Lemma 4.3(i) implies $\rho_\alpha = \mathrm{lca}(\alpha, \beta)$ and thus, similarly to case (i), $\beta \subseteq N^+(\alpha)$. On the other hand, by Lemma 4.3(ii) and $\rho_\alpha \succ \rho_\beta$, there is a leaf $x' \in L(T(\rho_\beta)) \setminus \alpha$ with $\sigma(x') = s$. Hence, $\mathrm{lca}(x', \beta) \prec \rho_\alpha = \mathrm{lca}(\alpha, \beta)$, which implies $\alpha \cap N^+(\beta) = \emptyset$.

(iii) The case $\rho_\alpha \prec \rho_\beta$ is symmetric to (ii).

(iv) If ρ_α, ρ_β are incomparable, it yields $\rho_\alpha, \rho_\beta \neq \rho$ and $\mathrm{lca}(\alpha, \beta) = \rho$, where ρ denotes the root of T. Since $\beta \preceq \rho_\beta$, Lemma 4.2 implies $\beta \cap N^+(\alpha) = \emptyset$. Similarly, $\alpha \cap N^+(\beta) = \emptyset$.

□

4.3.2 *Least Resolved Trees*

In general, there are many trees that explain the same 2-BMG. We next show that there is always a unique "smallest" tree among them, which we will call the least resolved tree for $(\vec{G}, \sigma)$. Later on, we will derive a hierarchy of leaf sets from $(\vec{G}, \sigma)$ whose tree representation coincides with this least resolved tree.

Recall that T_e is the tree obtained from T by contracting the inner edge $e = uv$. Analogously, we write T_A for the tree obtained by contracting all edges in A.

Definition 4.5. *Let $(\vec{G}, \sigma)$ be a BMG and (T, σ) a tree explaining $(\vec{G}, \sigma)$. An inner edge e in (T, σ) is* redundant *(w.r.t. $(\vec{G}, \sigma)$) if (T_e, σ) also explains $(\vec{G}, \sigma)$. Edges that are not redundant are called* relevant*.*

Note that for an outer edge $e = uv$, we have $v \notin L(T_e)$ and thus, (T_e, σ) does not explain $(\vec{G}, \sigma)$. In this chapter all redundant edges are redundant w.r.t. to some BMG, thus we omit the reference to the explicit graph whenever the context is clear.
The next two results characterize redundant edges and show that such edges can be contracted in an arbitrary order.

Lemma 4.5. *Let (T,σ) be a tree that explains a connected 2-BMG $(\vec{G},\sigma)$. Then the edge inner $e = uv$ is redundant if and only if there exists no* R*-class α such that $v = \rho_\alpha$.*

Proof. Suppose first that e is a redundant inner edge. Assume, for contradiction, that there is an R-class α such that $v = \rho_\alpha$. Since (T,σ) is phylogenetic, $L(T(u)) \setminus L(T(v))$ has to be non-empty. If there is a leaf $y \in L(T(u)) \setminus L(T(v))$ with $\sigma(y) \neq \sigma(\alpha)$ in (T,σ), then $y \notin N^+(\alpha)$ by Lemma 4.3(vi). But then contraction of e implies $y \in T(\rho_\alpha)$ and therefore $y \in N^+(\alpha)$, thus (T_e,σ) does not explain $(\vec{G},\sigma)$. Consequently, $L(T(u)) \setminus L(T(v))$ can only contain leaves x with $\sigma(x) = \sigma(\alpha)$. Indeed, for any $y' \in L(T(v))$ it is true that $v = \rho_\alpha = \mathrm{lca}(\alpha, y') \prec \mathrm{lca}(x, y')$, i.e., $N^-(x) \neq N^-(\alpha)$ and thus $x \notin \alpha$. By contracting e, we obtain $\mathrm{lca}(x,z) \succeq uv = \rho_\alpha$ which implies $N^+(x) = N^+(\alpha)$ and $N^-(x) = N^-(\alpha)$, and therefore $x \in \alpha$. Hence, (T_e,σ) does not explain $(\vec{G},\sigma)$; a contradiction.

Conversely, assume that there is no R-class α such that $v = \rho_\alpha$, i.e., for each $\alpha \in \mathcal{N}$ it either holds (i) $v \prec \rho_\alpha$, (ii) $v \succ \rho_\alpha$, or (iii) v and ρ_α are incomparable. In the first and second case, contraction of e implies either $v \preceq \rho_\alpha$ or $v \succeq \rho_\alpha$. Thus, since $L(T(w)) = L(T_e(w))$ is clearly satisfied if w and v are incomparable, we have $L(T(w)) = L(T_e(w))$ for every $w \neq v$. Moreover, $N^+(\alpha) = \{y \mid y \in L(T(\rho_\alpha)), \sigma(y) \neq \sigma(\alpha)\}$ by Lemma 4.3(vi). Together these facts imply for every R-class α with $\rho_\alpha \neq v$ that $N^+(\alpha)$ remains unchanged in (T_e,σ) after contraction of e. Since the out-neighborhoods of all R-classes are unaffected by contraction of e, all in-neighborhoods also remain the same in (T_e,σ). Therefore (T,σ) and (T_e,σ) explain the same graph $(\vec{G},\sigma)$. □

Lemma 4.6. *Let (T,σ) be a tree that explains a connected 2-BMG $(\vec{G},\sigma)$ and let e be a redundant edge. Then the edge $f \neq e$ is redundant in (T_e,σ) if and only if f is redundant in (T,σ). Moreover, if two edges $e \neq f$ are redundant in (T,σ), then $((T_e)_f,\sigma)$ also explains $(\vec{G},\sigma)$.*

Proof. Let $e = uv$ be a redundant edge in (T,σ). Then, for any vertex $w \neq u, v$ in (T,σ), it is true that w is the root of an R-class α in (T_e,σ) if and only if w is the root of α in (T,σ). In particular, the vertex uv in (T_e,σ) is the root of an R-class α' if and only if $u = \rho_{\alpha'}$ in (T,σ). Consequently, f is redundant in (T,σ) if and only if f is redundant in (T_e,σ). □

As an immediate consequence, contraction of edges is commutative, i.e., the order of the contractions is irrelevant. We can therefore write T_A for the tree obtained by contracting all edges in A in arbitrary order:

Corollary 4.2. *Let (T,σ) be a tree that explains a 2-BMG $(\vec{G},\sigma)$ and let A be a set of redundant edges of (T,σ). Then (T_A,σ) explains $(\vec{G},\sigma)$. In particular, $((T_A)_B,\sigma)$ explains $(\vec{G},\sigma)$ if and only if B is a set of redundant edges of (T,σ).*

This leads to the notion of so-called least resolved trees:

Definition 4.6. *Let $(\vec{G},\sigma)$ be a BMG explained by (T,σ). We say that (T,σ) is* least resolved *(w.r.t. $(\vec{G},\sigma)$) if (T_A,σ) does not explain $(\vec{G},\sigma)$ for any non-empty set A of inner edges of (T,σ).*

Again, we omit the explicit reference to the underlying BMG whenever the context is clear.
We are now in the position to formulate the main result of this section:

Theorem 4.2. *For any connected 2-BMG $(\vec{G}, \sigma)$, there exists a unique least resolved tree (T', σ) that explains $(\vec{G}, \sigma)$. (T', σ) is obtained by contraction of all redundant edges in an arbitrary tree (T, σ) explaining $(\vec{G}, \sigma)$. The set of all redundant edges in (T, σ) is given by*

$$\mathfrak{E}_T = \{e = uv \mid v \notin L(T) \text{ and there is no } \mathsf{R}\text{-class } \alpha \text{ such that } v = \rho_\alpha\}.$$

Moreover, (T', σ) is displayed by (T, σ).

Proof. Any edge in a least resolved tree (T', σ) is relevant and therefore, as a consequence of Cor. 4.2, (T', σ) is obtained from (T, σ) by contraction of all redundant edges of (T, σ). According to Lemma 4.5, the set of redundant edges is exactly $\mathfrak{E}_T$. Since the order of contracting the edges in $\mathfrak{E}_T$ is arbitrary, there is a least resolved tree for every given tree (T, σ).

Now assume, for contradiction, that there exist colored digraphs that are explained by two distinct least resolved trees. Let $(\vec{G}, \sigma)$ be a minimal graph (w.r.t. the number of vertices) that is explained by two distinct least resolved trees (T_1, σ) and (T_2, σ), and let $v \in L$ with $\sigma(v) = s$. By construction, the two trees (T_1', σ') and (T_2', σ') with $T_1' := T_{1|L\setminus\{v\}}$, $T_2' := T_{2|L\setminus\{v\}}$ and leaf labeling $\sigma' := \sigma_{|L\setminus\{v\}}$, each explain a unique graph, which we denote by $(\vec{G}_1', \sigma')$ and $(\vec{G}_2', \sigma')$, respectively. Lemma 4.1 implies that $(\vec{G}', \sigma') := (\vec{G}[L \setminus \{v\}], \sigma')$ is a subgraph of both $(\vec{G}_1', \sigma')$ and $(\vec{G}_2', \sigma')$.

We next show that $(\vec{G}_1', \sigma')$ and $(\vec{G}_2', \sigma')$ are equal by characterizing the additional edges that are inserted in both graphs compared to $(\vec{G}', \sigma')$. Assume that there is an additional edge (u, y) in one of the graphs, say $(\vec{G}_1', \sigma)$. Since (u, y) is not an edge in $(\vec{G}, \sigma)$, we have $\mathrm{lca}_T(u, y) \succ_T \mathrm{lca}_T(u, y')$ for some $y' \in L(T)$ with $\sigma(y) = \sigma(y')$. However, $(u, y) \in E(\vec{G}_1')$ implies that $\mathrm{lca}_{T_1}(u, y) \preceq_{T_1} \mathrm{lca}_{T_1}(u, y'')$ for all $y'' \in L \setminus \{v\}$ with $\sigma(y) = \sigma(y')$. Since $T_1' := T_1 \setminus \{v\}$, we obtain $\mathrm{lca}_T(u, y') \prec_T \mathrm{lca}_T(u, y) \preceq_T \mathrm{lca}_T(u, y'')$, which implies that $y' = v$ and, in particular, $(u, v) \in E(\vec{G})$ and $N^+(u) = \{v\}$.

In particular, we have $\sigma(u) = t \neq s$. In this case, u has no out-neighbors in $(\vec{G}', \sigma')$ but it has outgoing arcs in $(\vec{G}_1', \sigma')$ and $(\vec{G}_2', \sigma')$. In order to determine these outgoing arcs explicitly, we will reconstruct the local structure of (T_1, σ) and (T_2, σ) in the vicinity of the leaf v. The following argumentation is illustrated in Fig. 9.

Since $N^+(u) = \{v\}$, there is an R-class $\alpha = \{v\}$. Let β be the R-class of $(\vec{G}, \sigma)$ to which u belongs. It satisfies $N^+(\beta) = \{v\}$. Therefore $L(T_1(\rho_\beta)) \cap L[s] = \{v\}$ and $L(T_2(\rho_\beta)) \cap L[s] = \{v\}$. In particular, this implies $L(T_1(\rho_\alpha)) \cap L[s] = \{v\}$ and $L(T_2(\rho_\alpha)) \cap L[s] = \{v\}$. The children of ρ_α in both T_1 and T_2 must be leaves: otherwise, Lemma 4.3(ii) would imply that there are inner vertices $\rho_{\alpha'}$ and $\rho_{\beta'}$ below ρ_α, which in turn would contradict $L(T_1(\rho_\alpha)) \cap L[s] = \{v\}$ and $L(T_2(\rho_\alpha)) \cap L[s] = \{v\}$.

Moreover, the subtrees $T_1(\rho_\alpha)$ and $T_2(\rho_\alpha)$ must contain leaves of both colors. Thus there exists an R-class β' with color t whose root $\rho_{\beta'}$ coincides with ρ_α in

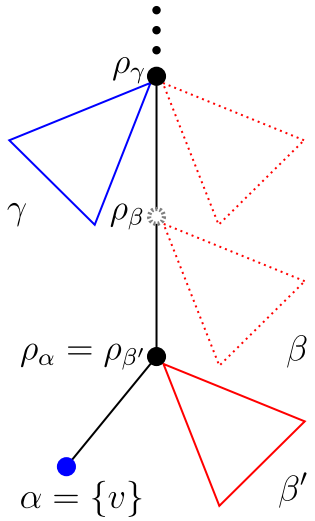


Fig. 9. Illustration of the proof of Thm. 4.2 showing the local subtrees of (T_1, σ) and (T_2, σ), immediately above $\alpha = \{v\}$. The relevant portion extends to the root ρ_γ of the R-class γ that is located immediately above α and has the same color as α, here blue. Clearly, the deletion of α can affect only pairs of vertices x, y with $\mathrm{lca}(x, y)$ below ρ_γ. Triangles denote the subtree that consists of all leaves of the corresponding class which are attached to the root of the class by an outer edge. Dashed triangles and nodes denote subtrees which may or may not be present in (T_1, σ) and (T_2, σ).

both (T_1, σ) and (T_2, σ). More precisely, we have $\mathsf{child}(\rho_\alpha) = \alpha \cup \beta'$. We now distinguish two cases:
(i) If $N^-(\beta) \cap \{v\} \neq \emptyset$ in $(\vec{G}, \sigma)$, we have $\rho_\beta = \rho_\alpha$, i.e., $\beta = \beta'$.
(ii) Otherwise if $N^-(\beta) \cap \{v\} = \emptyset$, then $\mathrm{lca}(v, \beta') \prec \mathrm{lca}(v, \beta)$, hence $\rho_\beta \succ \rho_\alpha$. In particular, since $N^+(\beta) = \{v\}$, Lemma 4.3(vi) implies that there cannot be any other R-class $\alpha' \neq \alpha$ of $(\vec{G}, \sigma)$ with color s and $\rho_\beta \succeq \rho_{\alpha'}$. Moreover, there cannot be any other class β'' of color t such that $\rho_{\beta''}$ is contained in the unique path from ρ_β to ρ_α, otherwise it holds $N^+(\beta'') = N^+(\beta)$ and $N^-(\beta'') = N^-(\beta)$ by Lemma 4.3(vi), i.e., $\beta'' \mathsf{R} \beta$. Therefore we conclude that $\rho_\beta \rho_\alpha \in E(T_1)$ as well as $\rho_\beta \rho_\alpha \in E(T_2)$.
If v is the only leaf of color s in $(\vec{G}, \sigma)$, it follows from (i) and (ii) that $(T_1', \sigma') = (T_1(\rho_\beta), \sigma') = (T_2(\rho_\beta), \sigma') = (T_2', \sigma')$; a contradiction. Hence, there is a unique tree representation for $(\vec{G}, \sigma)$ if $|L[s]| = 1$.
Now suppose $L[s] > 1$. Then, both in case (i) and case (ii) there is a parent of $\mathsf{par}(\rho_\beta)$ because otherwise $(\vec{G}_1', \sigma')$ and $(\vec{G}_2', \sigma')$ would not contain color s. In either case the parent of ρ_β is an inner node of the least resolved tree (T_1, σ') and (T_2, σ'), respectively. We claim that $\mathsf{par}(\rho_\beta)$ is the root of an R-class γ of color s. Suppose this is not the case, i.e., $\sigma(\gamma) = t$ and there is no other $\gamma' \in \mathcal{N}$ such that $\sigma(\gamma') = s$ and $\mathsf{par}(\rho_\beta) = \rho_{\gamma'}$. Then $N^+(\gamma) = N^+(\beta)$ and $N^-(\gamma) = N^-(\beta)$ by Lemma 4.3(vi), which implies that $\beta \mathsf{R} \gamma$ and ρ_β is not the root of β; a contradiction.

We therefore conclude that the local subtrees of (T_1, σ') and (T_2, σ') immediately above α, that is $(T_1(\rho_\gamma), \sigma'_{|L(T_1(\rho_\gamma))})$ and $(T_2(\rho_\gamma), \sigma'_{|L(T_2(\rho_\gamma))})$, as indicated in Fig. 9, are identical. Moreover, it follows $\mathrm{lca}(u, \gamma) \preceq \mathrm{lca}(u, w)$ for any $w \in L[s] \setminus \{v\}$. Hence, the additionally inserted edges in $(\vec{G}_1', \sigma)$ and $(\vec{G}_2', \sigma)$ are exactly the edges (u, c) for all $c \in \gamma$. We therefore conclude that $(\vec{G}_1', \sigma) = (\vec{G}_2', \sigma)$, which implies $(T_1', \sigma') = (T_2', \sigma')$. Since v has been chosen arbitrarily, this implies $(T_1, \sigma) = (T_2, \sigma)$; a contradiction. □

Finally, we consider a few simple properties of least resolved trees that will be useful in the following sections.

Corollary 4.3. *Let $(\vec{G}, \sigma)$ be a connected 2-BMG that is explained by a least resolved tree (T, σ). Then all elements of $\alpha \in \mathcal{N}$ are attached to ρ_α, i.e., $\rho_\alpha a \in E(T)$ for all $a \in \alpha$.*

Proof. Assume $\rho_\alpha a \notin E(T)$. Since by definition $\alpha \prec \rho_\alpha$, there exists an inner node v with $\rho_\alpha v \in E(T)$ such that v lies pn the unique path from ρ_α to a. In particular $v \neq a$. Thm. 4.2 implies that each inner vertex (except possibly the root) of the least resolved tree (T, σ) must be the root of some R-class of $(\vec{G}, \sigma)$. Hence, there is an R-class $\beta \in \mathcal{N}$ with $\rho_\beta = v$. According to Lemma 4.3(ii), the subtree $T(v)$ contains leaves of both colors, i.e., there exists some leaf $c \in L(T(v))$ with $\sigma(c) \neq \sigma(a)$. It follows $\mathrm{lca}(a, c) \prec \rho_\alpha$, which contradicts the definition of ρ_α.

□

This result remains true also for 2-BMGs that are not connected.

4.3.3 *Characterization of 2-BMGs*

We will first establish necessary conditions for a colored digraph to be a 2-BMG. The key construction for this purpose is the reachable set of an R-class, that is, the set of all leaves that can be reached from this class via a path of directed edges in $(\vec{G}, \sigma)$. Not unexpectedly, the reachable sets should form a hierarchical structure. However, this hierarchy does not quite determine a tree that explains $(\vec{G}, \sigma)$. We shall see, however, that the definition of reachable sets can be modified in such a way that the resulting hierarchy defines the unique least resolved tree w.r.t. $(\vec{G}, \sigma)$.

Necessary Conditions

We start by deriving some graph properties of 2-BMGs. We shall see later that these are in fact sufficient to characterize 2-BMGs.

Theorem 4.3. *Let $(\vec{G}, \sigma)$ be a connected 2-BMG. Then it holds for any two* R*-classes α and β of $(\vec{G}, \sigma)$:*

(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) $N^+(N^+(N^+(\alpha))) \subseteq N^+(\alpha)$.

(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* $N^+(\beta) \subseteq N^+(\alpha)$.

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

(N1): For $\sigma(\alpha) = \sigma(\beta)$ this is trivial, thus suppose $\sigma(\alpha) \neq \sigma(\beta)$. By Lemma 4.3(vi), α is not located in the subtree $T(\rho_\beta)$ and β is not located in the subtree $T(\rho_\alpha)$. Therefore ρ_α and ρ_β must be incomparable. Since $N^+(\alpha), N^+(N^+(\alpha)) \preceq \rho_\alpha$ and $N^+(\beta), N^+(N^+(\beta)) \preceq \rho_\beta$ by Lemma 4.3(iii) and (vii), we conclude that $N^+(\alpha) \cap N^+(N^+(\beta)) = N^+(\beta) \cap N^+(N^+(\alpha)) = \emptyset$.

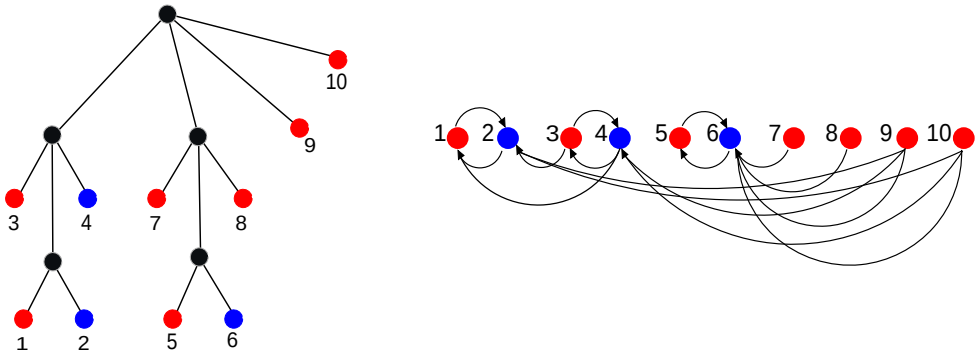


Fig. 10. A 2-BMG with $|\mathcal{W}| > 1$ and its least resolved tree. The R-class $\alpha = \{9, 10\}$ consists of children of the root without in-neighbors. There is a second R-class without in-neighbors, namely $\beta = \{7, 8\}$. Hence, $\mathcal{W} = \{\alpha, \beta\}$ and $R(\alpha) = \{1, \ldots, 6\} = L \setminus (\alpha \cup \beta)$, while $R(\beta) = \{5, 6\}$.

(N2): For contradiction, assume that there exists some $q \in N^+(N^+(N^+(\alpha))) \setminus N^+(\alpha)$. Since $\sigma(q) = \sigma(u) \neq \sigma(x)$ for all $x \in \alpha$ and $u \in N^+(\alpha)$, any such q must satisfy $\mathrm{lca}(x, q) \succ \mathrm{lca}(x, u)$ for all $x \in \alpha$ and $u \in N^+(\alpha)$. Otherwise it would be contained in $N^+(\alpha)$. Since $N^+(x) \preceq \rho_\alpha$ by Lemma 4.3(iii), the definition of ρ_α implies that there is some pair $x \in \alpha$ and $y \in \beta \subseteq N^+(\alpha)$ with $\mathrm{lca}(x, y) = \rho_\alpha$. Therefore $\mathrm{lca}(x, q) \succ \rho_\alpha$.

Now consider $\beta \subseteq N^+(\alpha)$. Since $\sigma(\beta) \neq \sigma(\alpha)$ and $\mathrm{lca}(\alpha, \beta) \preceq \rho_\alpha$, we infer that $N^+(N^+(\alpha)) \preceq \rho_\alpha$. Repeating the argument yields $N^+(N^+(N^+(\alpha))) \preceq \rho_\alpha$ and thus, there cannot be a pair of leaves $x \in \alpha$ and $q \in N^+(N^+(N^+(\alpha)))$ with $\mathrm{lca}(x, q) \succ \rho_\alpha$.

(N3): We first note that Property (N3) is trivially true for $\alpha = \beta$. Hence, assume $\alpha \neq \beta$ and suppose $N^+(\alpha) \cap N^+(\beta) \neq \emptyset$. Since T is a tree, Lemma 4.3(vi) implies that either $N^+(\alpha) \subseteq N^+(\beta)$ or $N^+(\beta) \subseteq N^+(\alpha)$. Assume $N^+(\beta) \subseteq N^+(\alpha)$. Hence, $\rho_\beta \preceq \rho_\alpha$. Consequently, for any $\gamma \subseteq N^-(\alpha)$ holds $\mathrm{lca}(\gamma, \beta) \preceq \mathrm{lca}(\gamma, \alpha) \preceq \mathrm{lca}(\gamma, x)$ for all x with $\sigma(x) = \sigma(\alpha)$ and therefore, $N^-(\alpha) \subseteq N^-(\beta)$. Assume, for contradiction, that there exists $\gamma' \subseteq N^-(\beta) \setminus N^-(\alpha)$. By definition, we have $\rho_\alpha \succeq \mathrm{lca}(\gamma', \beta) \succeq \rho_\beta$ in this case. But then Lemma 4.3(vi) implies $N^+(\gamma') \subseteq N^+(\alpha)$ and $\beta \subseteq N^+(\gamma') \subseteq N^+(N^+(\alpha))$; a contradiction. □

Definition 4.7. *For any properly colored digraph $(\vec{G}, \sigma)$ we define the* reachable set *$R(\alpha)$ for an* R*-class α by*

$$R(\alpha) = N^+(\alpha) \cup N^+(N^+(\alpha)) \cup N^+(N^+(N^+(\alpha))) \cup \ldots \tag{2}$$

Moreover, we write $\mathcal{W} := \{\alpha \in \mathcal{N} \mid N^-(\alpha) = \emptyset\}$ for the set of R*-classes without in-neighbors.*

As we shall see below, technical difficulties arise for distinct R-classes that share the same set of in-neighbors. Hence, we briefly consider the classes in $\mathcal{W}$. An example is shown Fig. 10.

Lemma 4.7. *Let $(\vec{G}, \sigma)$ be a connected 2-BMG explained by a tree (T, σ) with leaf set L. Then all* R*-classes in $\mathcal{W}$ have the same color and the cardinality of $\mathcal{W}$ distinguishes three types of roots as follows:*

(i) $\mathcal{W} = \emptyset$ if and only if $\rho_T = \rho_\alpha = \rho_\beta$ for two distinct R*-classes α and β.*

(ii) $|\mathcal{W}| > 1$ if and only if there is a unique R*-class $\alpha^* \in \mathcal{W}$ that is characterized by $R(\alpha^*) = L \setminus \bigcup_{\beta \in \mathcal{W}} \beta$. Furthermore, $\rho_{\alpha^*} = \rho_T$.*

(iii) If $\mathcal{W} = \{\alpha\}$, then $\rho_\alpha = \rho_T$ and $R(\alpha) = L \setminus \alpha$.

Proof. By Thm. 4.1, as $(\vec{G}, \sigma)$, is connected, there is at least one child v of the root ρ_T of T that itself is the root of a subtree with a single leaf color, i.e., $\sigma(L(T(v))) = \{s\}$. Assume, for contradiction, that there are two R-classes $\alpha, \beta \in \mathcal{W}$ with $s = \sigma(\alpha) \neq \sigma(\beta) = t$. Then, by definition, $\mathrm{lca}(v, x) = \rho_T$ for all $x \in \beta$, and furthermore, $(u, x) \in E(\vec{G})$ for all $u \in L(T(v))$. Since $x \in \beta$ has an in-arc, we have $\beta \notin \mathcal{W}$; a contradiction. All leaves in $\mathcal{W}$ therefore have the same color.

For the remainder of the proof we fix such a child v of the root ρ_T. By construction, all leaves below it belong to the same R-class, which we denote by $\omega = L(T(v))$. W.l.o.g. we assume $\sigma(v) = s$. Since $t \notin \sigma(L(T(v)))$, we have $\rho_\omega = \rho_T$ and thus, $N^+(\omega) = L[t]$ by Lemma 4.3(vi).

(i) Suppose $\mathcal{W} = \emptyset$. Then there exists $\beta \in \mathcal{N}_t$ such that $\beta \subseteq N^-(\omega)$. For each $b \in \beta$ we have $\mathrm{lca}(b, \omega) \preceq \mathrm{lca}(b, x)$ for all $x \in L[s]$. Since $\mathrm{lca}(b, \omega) = \rho_T$, we conclude $\rho_\beta = \rho_T = \rho_\omega$.

Conversely, suppose α and β are two distinct R-classes such that $\rho_\alpha = \rho_\beta = \rho_T$. By Lemma 4.3(v), $\sigma(\alpha) \neq \sigma(\beta)$. W.l.o.g. assume $\sigma(\alpha) = s$ and $\sigma(\beta) = t$. Since $L(T(\rho_\alpha)) = L(T(\rho_T)) = L$, Lemma 4.3(vi) implies $N^+(\alpha) = L[t]$ and $N^+(\beta) = L[s]$. Therefore $\alpha \in N^-(\gamma)$ for any $\gamma \in \mathcal{N}_t$ and $\beta \in N^-(\gamma)$ for any $\gamma \in \mathcal{N}_s$. Hence, $\mathcal{W} = \emptyset$.

(ii) If $\mathcal{W} \neq \emptyset$, Property (i) implies $\rho_\beta \neq \rho_T$ for all $\beta \in \mathcal{N}_t$ and thus, $\rho_\beta \prec \rho_T$. Hence, there is no $\beta \in \mathcal{N}_t$ with $\omega \subseteq N^+(\beta)$, i.e., $N^-(\omega) = \emptyset$, which implies $\omega \in \mathcal{W}$.

Consider $\gamma \in \mathcal{N}_s$. We have $N^-(\gamma) \neq \emptyset$ if and only if there exists $\zeta \in \mathcal{N}_t$ such that $\gamma \subseteq N^+(\zeta)$, i.e., if and only if $\gamma \subseteq N^+(L[t])$. Since $N^+(\omega) = L[t]$, we have $\gamma \notin \mathcal{W}$ if and only if $\gamma \subseteq N^+(N^+(\omega))$. In other words, $N^+(N^+(\omega)) = L[s] \setminus \bigcup_{\beta \in \mathcal{W}} \beta$. Using (N2), we obtain

$$R(\omega) = N^+(\omega) \cup N^+(N^+(\omega)) = L[t] \cup \bigcup \{\gamma \in \mathcal{N}_s | N^-(\gamma) \neq \emptyset\} = L \setminus \bigcup_{\gamma \in \mathcal{W}} \gamma \,.$$

Now suppose there is another $\alpha \in \mathcal{W}$ with $R(\alpha) = L \setminus \bigcup_{\gamma \in \mathcal{W}} \gamma$. We already know that $\sigma(\alpha) = s$ since all classes in $\mathcal{W}$ must have the same color. Hence, $L[t] \subseteq R(\alpha)$. Consequently, $\zeta \in N^+(\omega)$ if and only if $\zeta \in N^+(\alpha)$ and thus, $N^+(\alpha) = N^+(\omega)$. Since $\alpha, \omega \in \mathcal{W}$ implies $N^-(\alpha) = N^-(\omega) = \emptyset$, α and ω share both in- and out-neighbors and thus, $\alpha = \omega$. Therefore ω is unique.

(iii) From the proof of Property (ii), we know that $|\mathcal{W}| = 1$ implies that the unique member of $\mathcal{W}$ is ω. We already know that $\rho_\omega = \rho_T$.

□

Sufficient Conditions

We now turn to showing that the properties obtained in Thm. 4.3 are already sufficient for the characterization of 2-BMGs. To this end, we show that the extended reachable sets form a hierarchy whenever $(\vec{G}, \sigma)$ satisfies the Properties (N1), (N2), and (N3).
The following simple property we will be used throughout this section:

Lemma 4.8. *If $(\vec{G}, \sigma)$ is a connected properly 2-colored digraph satisfying* (N1)*, then it holds for any two* R*-classes α and β:*

$$N^+(\alpha) \cap N^+(\beta) = \emptyset \quad \textit{implies} \quad N^+(N^+(\alpha)) \cap N^+(N^+(\beta)) = \emptyset \tag{3}$$

If $(\vec{G}, \sigma)$ satisfies (N2)*, then $R(\alpha) = N^+(\alpha) \cup N^+(N^+(\alpha))$.*

Proof. For any $\gamma \subseteq N^+(\alpha)$ and any $\gamma' \subseteq N^+(\beta)$, (N1) implies $N^+(\gamma) \cap N^+(N^+(\beta)) = N^+(\gamma') \cap N^+(N^+(\alpha)) = \emptyset$. Recall that (N0) holds by definition of R-classes. Hence, $N^+(\alpha)$ is the disjoint union of R-classes, i.e., $N^+(\alpha) = \bigcup_{\gamma \subseteq N^+(\alpha)} \gamma$. Thus $N^+(N^+(\alpha)) \cap N^+(N^+(\beta)) = (\bigcup_{\gamma \subseteq N^+(\alpha)} N^+(\gamma)) \cap N^+(N^+(\beta)) = \emptyset$. The equation $R(\alpha) = N^+(\alpha) \cup N^+(N^+(\alpha))$ is an immediate consequence of (N2). □

Lemma 4.9. *Let $(\vec{G}, \sigma)$ be a connected properly 2-colored digraph satisfying Properties* (N1)*,* (N2)*, and* (N3)*. Then $\mathcal{H} := \{R(\alpha) \mid \alpha \in \mathcal{N}\}$ is a hierarchy on $L \setminus \bigcup_{\alpha \in \mathcal{W}} \alpha$.*

Proof. First we note that $R(\alpha) = N^+(\alpha) \cup N^+(N^+(\alpha))$ by Property (N2). Furthermore, using (N0), we observe that $\beta \cap N^+(\alpha) \neq \emptyset$ implies $\beta \subseteq N^+(\alpha)$ for all R-classes α and β. In particular, therefore, $N^+(\alpha)$ is a disjoint union of R-classes, and thus $N^+(N^+(\alpha)) = \bigcup_{\beta \subseteq N^+(\alpha)} N^+(\beta)$ is again a disjoint union of R-classes. Hence, for any R-class $\beta \neq \alpha$, we either have $\beta \subseteq R(\alpha)$ or $\beta \cap R(\alpha) = \emptyset$. Note that the case $\alpha = \beta$ is trivial.

Suppose first $\beta \subseteq R(\alpha)$. If $\beta \subseteq N^+(\alpha)$, then $R(\beta) = N^+(\beta) \cup N^+(N^+(\beta)) \subseteq N^+(N^+(\alpha)) \cup N^+(N^+(N^+(\alpha))) \subseteq N^+(N^+(\alpha)) \cup N^+(\alpha)$. On the other hand, $\beta \subseteq N^+(N^+(\alpha))$ yields $R(\beta) \subseteq N^+(N^+(N^+(\alpha))) \cup N^+(N^+(N^+(N^+(\alpha)))) \subseteq N^+(\alpha) \cup N^+(N^+(\alpha))$. Thus, $R(\beta) \subseteq R(\alpha)$.
Exchanging the roles of α and β, the same argument shows that $\alpha \subseteq R(\beta)$ implies $R(\alpha) \subseteq R(\beta)$.

Now suppose that neither $\alpha \subseteq R(\beta)$ nor $\beta \subseteq R(\alpha)$ is satisfied and thus, by the arguments above, that $\alpha \cap R(\beta) = \beta \cap R(\alpha) = \emptyset$. In particular, therefore, $\alpha \cap N^+(\beta) = \beta \cap N^+(\alpha) = \emptyset$ and thus, Property (N1) implies $R(\alpha) \cap R(\beta) = (N^+(\alpha) \cap N^+(\beta)) \cup (N^+(N^+(\alpha)) \cap N^+(N^+(\beta)))$. If $N^+(\alpha) \cap N^+(\beta) = \emptyset$, then $R(\alpha) \cap R(\beta) = \emptyset$ by Lemma 4.8. If $N^+(\alpha) \cap N^+(\beta) \neq \emptyset$, then Property (N3) and $\alpha \cap R(\beta) = \beta \cap R(\alpha) = \emptyset$ implies either $N^+(\alpha) \subseteq N^+(\beta)$ or $N^+(\beta) \subseteq N^+(\alpha)$. Isotony of N^+ thus implies $N^+(N^+(\alpha)) \subseteq N^+(N^+(\beta))$ or $N^+(N^+(\beta)) \subseteq N^+(N^+(\alpha))$, respectively. Hence, we have either $R(\alpha) \subseteq R(\beta)$ or $R(\beta) \subseteq R(\alpha)$. Therefore $\mathcal{H}$ is a hierarchy.

Finally, we proceed to show that there is a unique set $R(\alpha^*)$ that is maximal w.r.t. inclusion and, in particular, satisfies $R(\alpha^*) = L \setminus \bigcup_{\alpha \in \mathcal{W}} \alpha$.
Assume, for contradiction, that there are two distinct elements $R(\alpha), R(\alpha^*) \in$

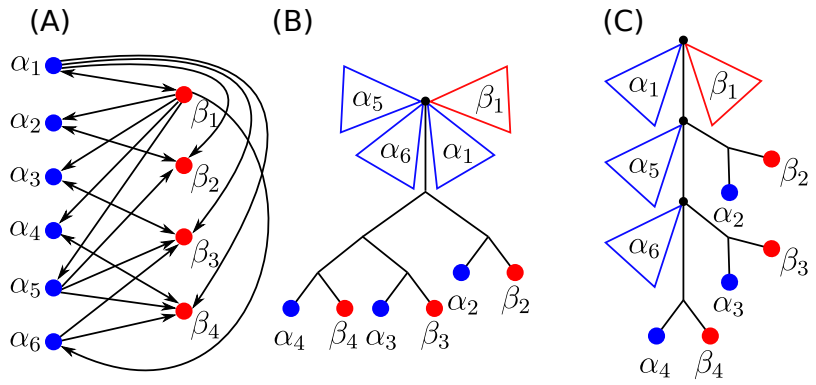


Fig. 11. (A) The properly 2-colored digraph $(\vec{G}, \sigma)$ satisfies (N1), (N2), and (N3). All α_i are R-classes of $(\vec{G}, \sigma)$ and belong to color "blue", the R-classes β_j form the "red" color classes. Red (blue) triangles indicate subtrees that only contain red (blue) leaves. Note that $N^-(\alpha_1) = N^-(\alpha_5) = N^-(\alpha_6)$. (B) The tree obtained from the hierarchy $\mathcal{H} = \{R(\alpha) \mid \alpha \in \mathcal{N}\}$ by attaching to the corresponding tree the elements of α as leaves to $R(\alpha)$, does not explain $(\vec{G}, \sigma)$. It would imply $N^-(\alpha_1) = N^-(\alpha_5) = N^-(\alpha_6)$ and $N^+(\alpha_1) = N^+(\alpha_5) = N^+(\alpha_6)$, i.e., $\alpha_1 \mathsf{R} \alpha_5 \mathsf{R} \alpha_6$. (C) The tree defined by the hierarchy $\mathcal{H}' = \{R'(\alpha) \mid \alpha \in \mathcal{N}\}$ with elements of α attached as leaves to $R'(\alpha)$ is the unique least resolved tree that explains $\vec{G}$ (cf. Lemma 4.11).

$\mathcal{H}$ that are both maximal w.r.t. inclusion. Thus $R(\alpha) \cap R(\alpha^*) = \emptyset$ and $\alpha \neq \alpha^*$. Moreover, since $\mathcal{H}$ is a hierarchy, we must have $R(\beta) \cap R(\alpha^*) = \emptyset$ for each $\beta \in \mathcal{N}$ with $R(\beta) \subseteq R(\alpha)$. In particular, this implies $\beta \subseteq R(\alpha)$ for any $\beta \in \mathcal{N}$ with $R(\beta) \subseteq R(\alpha)$. As a consequence there is no $\beta \subseteq R(\alpha)$ and $\beta' \subseteq R(\alpha^*)$ such that $\beta \subseteq N^+(\alpha^*)$ and $\beta' \subseteq N^+(\alpha)$, respectively. Therefore $R(\alpha)$ and $R(\alpha^*)$ are not connected; a contradiction to the connectedness of $\vec{G}$. Hence, $R(\alpha) = R(\alpha^*)$, i.e., there is a unique set $R(\alpha^*)$ in $\mathcal{H}$ that is maximal w.r.t. inclusion. It contains all R-classes of $\vec{G}$ that have non-empty in-neighborhood. Since, by definition, all vertices of $\vec{G}$ are assigned to exactly one R-class, we conclude that $R(\alpha^*) = L \setminus \bigcup_{\alpha \in \mathcal{W}} \alpha$. □

Note that while $R(\alpha)$ is unique for a given R-class α, there may exist more than one R-class that have the same reachable set (see for instance α_2 and β_2 in Fig. 11(C)). In particular, there may even be R-classes with different color giving rise to the same element of $\mathcal{H}$. More generally, we have $R(\alpha) = R(\beta)$ for $\alpha \neq \beta$ if and only if $\alpha \in R(\beta)$ and $\beta \in R(\alpha)$.

A hierarchy $\mathcal{H}$ corresponds to a unique tree $T(\mathcal{H})$ defined as the Hasse diagram of $\mathcal{H}$, i.e., the vertices of $T(\mathcal{H})$ are sets of $\mathcal{H}$, and R_2 is a child of R_1 if and only if $R_2 \subset R_1$ and there is no R_3 such that $R_2 \subset R_3 \subset R_1$. In particular, thus, two R-classes belong to the same inner vertex if $R(\alpha) = R(\beta)$. It is tempting to use this tree to construct a tree T explaining $(\vec{G}, \sigma)$ by attaching the elements of α as leaves to the node $R(\alpha)$ in $T(\mathcal{H})$. The example in Fig. 11 shows, however, that this simply does not work. The key issue arises from groups of distinct R-classes that share the same in-neighborhood because they will in general be attached to the same node in $T(\mathcal{H})$, i.e., they are indistinguishable. We therefore need a modification of the definition of reachable sets that properly distinguishes such R-classes in order to construct a hierarchy with

the appropriate resolution for the least resolved tree specified in Thm. 4.2. To this end, we define for every R-class of a properly 2-colored digraph $(\vec{G}, \sigma)$ the auxiliary leaf set

$$Q(\alpha) = \{x \in V(\vec{G}) \mid \exists \beta \in \mathcal{N} : x \in \beta,\, N^-(\beta) = N^-(\alpha) \text{ and } N^+(\beta) \subseteq N^+(\alpha)\} \tag{4}$$

Note that $\alpha \subseteq Q(\alpha)$. For later reference we list several simple properties of Q.

Lemma 4.10. *Let $(\vec{G}, \sigma)$ be a 2-BMG and $\alpha, \beta \in \mathcal{N}$. Then*

(i) $\beta \subseteq Q(\alpha)$ *implies* $\sigma(\beta) = \sigma(\alpha)$,

(ii) $\beta \subseteq Q(\alpha)$ *implies* $Q(\beta) \subseteq Q(\alpha)$,

(iii) $\beta \subseteq Q(\alpha)$ *implies* $R(\beta) \subseteq R(\alpha)$,

(iv) $\alpha \cap N^+(\beta) = \emptyset$ *implies* $Q(\alpha) \cap N^+(\beta) = \emptyset$, *and*

(v) $\alpha \cap N^+(N^+(\beta)) = \emptyset$ *implies* $Q(\alpha) \cap N^+(N^+(\beta)) = \emptyset$.

Proof. (i) follows directly from the definition.

(ii) Let $\beta \subseteq Q(\alpha)$, $\gamma \in \mathcal{N}$, and $\gamma \subseteq Q(\beta)$. Then $N^-(\gamma) = N^-(\beta) = N^-(\alpha)$ and $N^+(\gamma) \subseteq N^+(\beta) \subseteq N^+(\alpha)$. Hence, $\gamma \subseteq Q(\alpha)$ and therefore, $Q(\beta) \subseteq Q(\alpha)$.

(iii) By definition, $N^+(\beta) \subseteq N^+(\alpha)$. Monotonicity of N^+ implies $N^+(N^+(\beta)) \subseteq N^+(N^+(\alpha))$ and therefore, $R(\beta) \subseteq R(\alpha)$.

(iv) Assume that $\alpha \cap N^+(\beta) = \emptyset$ but $\gamma \subseteq Q(\alpha) \cap N^+(\beta) \neq \emptyset$. Thus $\beta \subseteq N^-(\gamma) = N^-(\alpha)$, i.e., $\alpha \subseteq N^+(\beta)$; a contradiction.

(v) Assume that $\alpha \cap N^+(N^+(\beta)) = \emptyset$ but $\gamma \subseteq Q(\alpha) \cap N^+(N^+(\beta)) \neq \emptyset$. Thus there is an R-class $\xi \subseteq N^+(\beta)$ such that $\xi \subseteq N^-(\gamma) = N^-(\alpha)$ and therefore, $\alpha \subseteq N^+(N^+(\beta))$; a contradiction.

□

Finally we define, for any R-class in a properly 2-colored digraph $(\vec{G}, \sigma)$, its *extended reachable set* as

$$R'(\alpha) := R(\alpha) \cup Q(\alpha). \tag{5}$$

Note that $\alpha \in R'(\alpha)$. Furthermore, the extended reachable set $R'(\alpha)$ contains vertices with both colors for every R-class α. Thus $|R'(\alpha)| > 1$. We show next that for any 2-BMG the extended reachable sets form the hierarchy that yields the desired least resolved tree.

Lemma 4.11. *Let $(\vec{G}, \sigma)$ be a connected properly 2-colored digraph satisfying Properties* (N1), (N2), *and* (N3). *Then $\mathcal{H}' := \{R'(\alpha) \mid \alpha \in \mathcal{N}\}$ is a hierarchy on L.*

Proof. Consider two distinct R-classes $\alpha, \beta \in \mathcal{N}$. By definition, $Q(\alpha)$ is the disjoint union of R-classes. The same is true for $R(\alpha)$ as argued in the proof of Lemma 4.9, hence $R'(\alpha) = R(\alpha) \cup Q(\alpha)$ is also the disjoint union of R-classes. Thus we have either $\beta \subseteq R'(\alpha)$ or $\beta \cap R'(\alpha) = \emptyset$.

First assume $\beta \subseteq R'(\alpha)$. Thus we have $\beta \subseteq R(\alpha)$ or $\beta \subseteq Q(\alpha)$. If $\beta \subseteq Q(\alpha)$, i.e., $N^+(\beta) \subseteq N^+(\alpha)$ and consequently $R(\beta) \subseteq R(\alpha)$, then Lemma 4.10(ii)+(iii) implies $R'(\beta) \subseteq R'(\alpha)$. If $\beta \subseteq R(\alpha)$, then $R(\beta) \subseteq R(\alpha) \subseteq R'(\alpha)$, which can be shown as in the proof of Lemma 4.9. It remains to show $Q(\beta) \subseteq R'(\alpha)$. By definition, we have $N^-(\gamma) = N^-(\beta)$ for any $\gamma \subseteq Q(\beta)$. Therefore $\beta \subseteq N^+(\alpha) \cup N^+(N^+(\alpha))$ implies $\gamma \subseteq N^+(\alpha) \cup N^+(N^+(\alpha))$. Hence, $\gamma \subseteq R(\alpha) \subseteq R'(\alpha)$. In summary, we have $R'(\beta) \subseteq R'(\alpha)$ for all $\beta \subseteq R'(\alpha)$.

The implication "$\alpha \subseteq R'(\beta) \implies R'(\alpha) \subseteq R'(\beta)$" follows by exchanging α and β in the previous paragraph.

Now suppose $\beta \cap R'(\alpha) = \alpha \cap R'(\beta) = \emptyset$. In particular, it then holds $\alpha \cap N^+(\beta) = \beta \cap N^+(\alpha) = \emptyset$ and $\alpha \cap N^+(N^+(\beta)) = \beta \cap N^+(N^+(\alpha)) = \emptyset$. Applying Property (N1) and Lemma 4.10(iv)+(v) yields $R'(\alpha) \cap R'(\beta) = \Big(N^+(\alpha) \cap N^+(\beta)\Big) \cup \Big(N^+(N^+(\alpha)) \cap N^+(N^+(\beta))\Big) \cup \Big(Q(\alpha) \cap Q(\beta)\Big)$. First, let $N^+(\alpha) \cap N^+(\beta) = \emptyset$. This immediately implies $Q(\alpha) \cap Q(\beta) = \emptyset$ and from Lemma 4.8 follows $N^+(N^+(\alpha)) \cap N^+(N^+(\beta)) = \emptyset$. Hence, $R'(\alpha) \cap R'(\beta) = \emptyset$. Now assume $N^+(\alpha) \cap N^+(\beta) \neq \emptyset$. By Property (N3), we conclude $N^-(\alpha) = N^-(\beta)$ and either $N^+(\alpha) \subseteq N^+(\beta)$ or $N^+(\beta) \subseteq N^+(\alpha)$. Consequently, either $N^+(N^+(\alpha)) \subseteq N^+(N^+(\beta))$ and $Q(\alpha) \subseteq Q(\beta)$, or $N^+(N^+(\beta)) \subseteq N^+(N^+(\alpha))$ and $Q(\beta) \subseteq Q(\alpha)$. Hence, it must either hold $R'(\alpha) \subseteq R'(\beta)$ or $R'(\beta) \subseteq R'(\alpha)$.

It remains to show that $L \in \mathcal{H}'$. Similar arguments as in the proof of Lemma 4.9 can be applied in order to show that there is a unique element $R'(\alpha^*)$ that is maximal w.r.t. inclusion in $\mathcal{H}'$. Since for any $\alpha \in \mathcal{N}$ it is true that $\alpha \in R'(\alpha)$, every R-class of $\vec{G}$ is contained in at least one element of $\mathcal{H}'$. Moreover, any vertex of $(\vec{G}, \sigma)$ is contained in exactly one R-class. Hence, $L = R'(\alpha^*) \in \mathcal{H}'$.

□

Since $\mathcal{H}'$ is a hierarchy, its Hasse diagram is a tree $T(\mathcal{H}')$. Its vertices are by construction exactly the extended reachable sets $R'(\alpha)$ of $(\vec{G}, \sigma)$. Starting from $T(\mathcal{H}')$, we construct the tree $T^*(\mathcal{H}')$ by attaching the vertices $x \in \alpha$ to the vertex $R'(\alpha)$ of $T(\mathcal{H}')$. The tree $T^*(\mathcal{H}')$ has leaf set L. Since $|R'(\alpha)| > 1$ as noted below Equ. (5), $T^*(\mathcal{H}')$ is a phylogenetic tree.

Theorem 4.4. *Let $(\vec{G}, \sigma)$ be a connected properly 2-colored digraph. Then there exists a tree T explaining $(\vec{G}, \sigma)$ if and only if $(\vec{G}, \sigma)$ satisfies Properties* (N1), (N2), *and* (N3). *The tree $T^*(\mathcal{H}')$ is the unique least resolved tree that explains $(\vec{G}, \sigma)$.*

Proof. The "only if"-direction is an immediate consequence of Lemma 4.2 and Thm. 4.3. For the "if"-direction we employ Lemma 4.11 and show that the tree $T^*(\mathcal{H}')$ constructed from the hierarchy $\mathcal{H}'$ explains $(\vec{G}, \sigma)$.

Let $x \in V(\vec{G})$ and α be the R-class of $(\vec{G}, \sigma)$ to which x belongs. Denote by $\tilde{N}^+(x)$ the out-neighbors of x in the graph explained by $T^*(\mathcal{H}')$. Therefore $y \in \tilde{N}^+(x)$ if and only if $\sigma(y) \neq \sigma(x)$ and $\mathrm{lca}_{T^*(\mathcal{H}')}(x, y)$ is the inner node to which x is attached in $T(\mathcal{H}')$, i.e., $R'(\alpha)$. Therefore $y \in \tilde{N}^+(x)$ if and only if $\sigma(y) \neq \sigma(x)$ and $y \in R'(\alpha)$. By (N2), this is the case if and only if $y \in N^+(x)$. Thus $\tilde{N}^+(x) = N^+(x)$. Since two digraphs are identical whenever all their out-neighborhoods are the same, the tree $T^*(\mathcal{H}')$ indeed explains $(\vec{G}, \sigma)$.
By construction and Thm. 4.2, $(T^*(\mathcal{H}'), \sigma)$ is a least resolved tree. □

4.3.4 *Informative Triples*

An inspection of induced subgraphs on three vertice of a 2-BMG $(\vec{G}, \sigma)$ shows that several local configurations derive only from specific types of trees. More precisely, certain induced subgraphs on three vertices are associated with uniquely defined triples displayed by the least resolved tree (T, σ) introduced in the previous section. Other induced subgraphs on three vertices, however, may derive from two or three distinct triples. The importance of triples derives from the fact that a phylogenetic tree can be reconstructed from the triples that it displays by the `BUILD` algorithm (see Section 3.4).

It is natural to ask whether the triples that can be inferred directly from $(\vec{G}, \sigma)$ are sufficient to (a) characterize 2-BMGs and (b) completely determine the least resolved tree (T, σ) explaining $(\vec{G}, \sigma)$.

Definition 4.8. *Let $(\vec{G}, \sigma)$ be a properly 2-colored digraph. We say that a triple $ab|c$ is* informative *(for $(\vec{G}, \sigma)$) if the three distinct vertices $a, b, c \in V(\vec{G})$ induce a colored subgraph $\vec{G}[a, b, c]$ isomorphic to the graphs X_1, X_2, X_3, or X_4 shown in Fig. 12. The set of informative triples is denoted by $\mathcal{R}(\vec{G}, \sigma)$.*

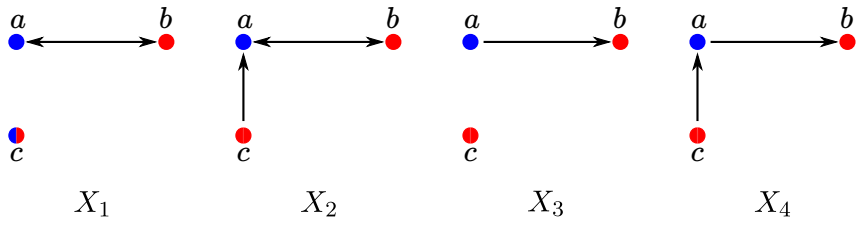


Fig. 12. Each of the three-vertex induced subgraphs X_1, X_2, X_3, and X_4 gives a triple $ab|c$. If vertex c in the drawing has two colors, then the color $\sigma(c)$ does not matter.

Lemma 4.12. *If $(\vec{G}, \sigma)$ is a connected 2-BMG, then each triple in $\mathcal{R}(\vec{G}, \sigma)$ is displayed by any tree (T, σ) that explains $(\vec{G}, \sigma)$.*

Proof. Let (T, σ) be a tree that explains $(\vec{G}, \sigma)$. Assume that there is an induced subgraph X_1 in $(\vec{G}, \sigma)$. W.l.o.g. let $\sigma(c) = \sigma(b)$. Since there is no arc (a, c) but an arc (a, b), we have $\mathrm{lca}(a, b) \prec \mathrm{lca}(a, c)$, which implies that T must display the triple $ab|c$. By the same arguments, if X_2, X_3, or X_4 is an induced subgraph in $(\vec{G}, \sigma)$, then T must display the triple $ab|c$. □

In particular, therefore, if $(\vec{G}, \sigma)$ is a 2-BMG, then $\mathcal{R}(\vec{G}, \sigma)$ is consistent. It is tempting to conjecture that consistency of the set $\mathcal{R}(\vec{G}, \sigma)$ of informative triples is already sufficient to characterize a 2-BMG. The example in Fig. 13 shows, however, that this is not the case.

Lemma 4.13. *Let (T, σ) be a least resolved tree explaining a connected 2-BMG $(\vec{G}, \sigma)$. Then every inner edge of T is distinguished by at least one triple in $\mathcal{R}(\vec{G}, \sigma)$.*

Proof. Let $e = uv$ be an inner edge of T. Since (T, σ) is least resolved for $(\vec{G}, \sigma)$, Thm. 4.2 implies that the edge e is relevant and hence, there exists a,

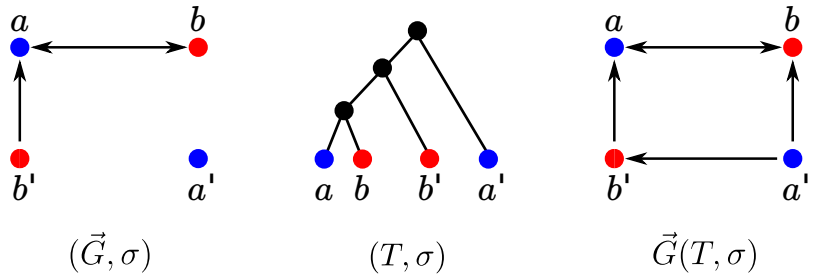


Fig. 13. The 4-vertex graph $(\vec{G},\sigma)$ on the l.h.s. cannot be a 2-BMG because there is no out-arc from a'. The four induced subgraphs are of type X_1, X_2, X_3 (with red and blue exchanged) and arc-less, respectively, resulting in the set $R(\vec{G},\sigma) = \{ab|b', ab|a', ab'|a'\}$ of informative triples. This set is consistent and displayed by the Aho tree T shown in the middle. It is not difficult to check that every edge of T is distinguished by one informative triple. Therefore $R(\vec{G},\sigma)$ identifies the leaf-colored tree (T,σ) [84]. However, the graph $\vec{G}(T,\sigma)$ explained by the tree (T,σ) is not isomorphic to the graph $(\vec{G},\sigma)$ from which the triples were inferred.

R-class $\alpha \in \mathcal{N}$ such that $v = \rho_\alpha$. By Cor. 4.3, we have $a \in \mathsf{child}(v)$ for any $a \in \alpha$. Lemma 4.3(ii) implies that $T(v)$ contains an R-class β with $\sigma(\alpha) \neq \sigma(\beta)$ and $b \in \beta$.

Case A: Suppose that $\rho_\beta = \rho_\alpha$ and therefore, $(a,b),(b,a) \in E(\vec{G})$. If u is the root of some R-class with $c \in \gamma$, then Lemma 4.3(vi) implies $(c,a) \in E(\vec{G})$, $(c,b) \notin E(\vec{G})$ for $\sigma(c) = \sigma(b)$ and $(c,b) \in E(\vec{G})$, $(c,a) \notin E(\vec{G})$ for $\sigma(c) = \sigma(a)$. In all cases, we neither have $(b,c) \in E(\vec{G})$ nor $(a,c) \in E(\vec{G})$ since $(a,b),(b,a) \in E(\vec{G})$. Therefore we always obtain a 3-vertex induced subgraph that is isomorphic to X_2 (see Fig. 12) and $ab|c \in \mathcal{R}(\vec{G},\sigma)$. On the other hand, if there is no R-class γ such that $u = \rho_\gamma$, then u is the root of (T,σ) by Cor. 4.3. Since (T,σ) is phylogenetic and u is not the root of any R-class, there must be an inner vertex $w \in \mathsf{child}(u) \setminus \{v\}$ such that $w = \rho_\gamma$ for some $\gamma \in \mathcal{N}$. Since $T(\rho_\gamma)$ contains leaves of both colors by Lemma 4.3(ii), for any leaf $c \in L(T(\rho_\gamma))$ there is no edge between c and b as well as between c and a. Taken together, we obtain the induced subgraph X_1 and the triple $ab|c$.

Case B: Now assume $\rho_\beta \prec \rho_\alpha$ and there is no other $\beta' \in \mathcal{N}$ with $\sigma(\beta') = \sigma(\beta)$ and $\rho_\alpha = \rho_{\beta'}$. By definition of ρ_β, we have $\mathrm{lca}(b,a') \prec \mathrm{lca}(b,a)$ for some a' with $\sigma(a) = \sigma(a')$, i.e., $(b,a) \notin E(\vec{G})$. Moreover, Lemma 4.3(vi) implies $b \in N^+(a)$, thus $(a,b) \in E(\vec{G})$. Similar to Case A, first suppose that u is the root of some R-class of $(\vec{G},\sigma)$. Since e is relevant, there is a $\gamma \in \mathcal{N}$ with $u = \rho_\gamma$ and $\sigma(\gamma) \neq \sigma(\alpha)$. Otherwise, if $\sigma(\gamma) = \sigma(\alpha)$ and there is no other $\gamma' \in \mathcal{N}$ with $u = \rho_{\gamma'}$, Lemma 4.3(vi) implies $N^+(\alpha) = N^+(\gamma)$ and $N^-(\alpha) = N^-(\gamma)$, i.e., α and γ belong to the same R-class with root u. Hence, v is not the root of any R-class; a contradiction. Consequently, we have $\sigma(\gamma) \neq \sigma(\alpha)$, thus $(c,a) \in E(\vec{G})$ by Lemma 4.3(vi) but $(a,c) \notin E(\vec{G})$. This yields the triple $ab|c$ that is derived from the subgraph X_4. If u is no root of any R-class, analogous arguments as in *Case A* show that there is an inner vertex $w \in \mathsf{child}(u) \setminus \{v\}$ such that the tree $T(w)$ contains leaves of both colors. In particular, there exists a leaf $c \in L(T(w))$ and since u is not the root of α, β or the R-class that c belongs

to, there is no arc between c and a or b in $(\vec{G}, \sigma)$. Hence, we again obtain the triple $ab|c$ which in this case is derived from X_3.

In every case we have $v = \mathrm{lca}(a, b) \prec \mathrm{lca}(a, c) = u$, i.e., the triple $ab|c$ distinguishes uv.

□

Lemma 4.13 suggests that the leaf-colored Aho tree $(\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma)), \sigma)$ of the set of informative triples $\mathcal{R}(\vec{G}, \sigma)$ explains a given 2-BMG $(\vec{G}, \sigma)$. The following result shows that this is indeed the case and sets the stage for one of the main results of this section, a characterization of 2-BMGs in terms of informative triples.

Theorem 4.5. *Let $(\vec{G}, \sigma)$ be a connected 2-BMG. Then $(\vec{G}, \sigma)$ is explained by the Aho tree of the set of informative triples, i.e., $(\vec{G}, \sigma) = \vec{G}(\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma)), \sigma)$.*

Proof. Let $(\tilde{T}, \sigma)$ be the unique least resolved tree that explains $(\vec{G}, \sigma)$ and set $L = V(\vec{G})$. For a fixed vertex $v \in L$ we write $(\vec{G}', \sigma') = (\vec{G}[L \setminus \{v\}], \sigma_{|L\setminus\{v\}})$. Let $(\tilde{T}', \sigma')$ be the unique least resolved tree that explains $(\vec{G}', \sigma')$ and let $(T', \sigma') := (\mathrm{Aho}(\mathcal{R}(\vec{G}', \sigma')), \sigma')$ be the leaf-colored Aho tree of the informative triples of $(\vec{G}', \sigma')$.

First consider the case $L = \{x, y\}$. Since $(\vec{G}, \sigma)$ is a connected 2-BMG, we have $\sigma(x) \neq \sigma(y)$ and $(x, y), (y, x) \in E(\vec{G})$. It is easy to see that both the least resolved tree w.r.t. $(\vec{G}, \sigma)$ and $\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma))$ correspond to the path $x - \rho_T - y$ with end points x and y. Thus $(\vec{G}, \sigma) = \vec{G}(\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma)), \sigma)$.

Now let $|L| > 2$ and assume, for contradiction, that the statement of the proposition is false. Then there is a minimal graph $(\vec{G}, \sigma)$ such that $(\vec{G}, \sigma) \neq \vec{G}(T, \sigma)$ for $(T, \sigma) = (\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma)), \sigma)$, i.e., $(\vec{G}', \sigma') = \vec{G}(T', \sigma')$ holds for every choice of $v \in V(\vec{G})$. Since $(\vec{G}, \sigma)$ is connected, Thm. 4.1 implies that there is an R-class α of $(\vec{G}, \sigma)$ such that $\rho_\alpha = \rho_{\tilde{T}}$. We fix a vertex v in this class α and proceed to show that $(\vec{G}, \sigma) = \vec{G}(T, \sigma)$; a contradiction. Let $\sigma(\alpha) = s$ and let $(\tilde{T} - v, \sigma')$ be the tree that is obtained by removing the leaf v and its incident edge from $(\tilde{T}, \sigma)$. Clearly, the out-neighborhood of every leaf of color s is still the same in $(\tilde{T} - v, \sigma')$ compared to $(\tilde{T}, \sigma)$. Moreover, Lemma 4.3(vi) implies that $N^+(x)$ remains unchanged in $(\tilde{T} - v, \sigma')$ for any $x \in L[t] \setminus \{v\}$ that belongs to an R-class β with $\rho_\beta \neq \rho_{\tilde{T}}$. If $\rho_\beta = \rho_{\tilde{T}}$, then $N^+(x) = L[s]$ in $(\tilde{T}, \sigma)$ by Lemma 4.3(vi) and thus, $N^+(x) = L[s] \setminus \{v\}$ in $(\tilde{T} - v, \sigma')$. We can therefore conclude that $(\tilde{T} - v, \sigma')$ explains the induced subgraph $(\vec{G}', \sigma')$ of $(\vec{G}, \sigma)$.

Now we distinguish two cases:

Case A: Let $|\mathsf{child}(\rho_{\tilde{T}}) \cap L| > 1$, which implies $|\mathsf{child}(\rho_{\tilde{T}-v}) \cap L| \geq 1$. Hence, the root of $(\tilde{T} - v, \sigma')$ has at least two children, which in particular implies that $\vec{G}(\tilde{T} - v, \sigma')$ is connected by Thm. 4.1. Since $(\tilde{T}, \sigma)$ is least resolved, Thm. 4.2 implies that any inner edge of $(\tilde{T} - v, \sigma')$ is relevant and hence, $(\tilde{T}', \sigma') = (\tilde{T} - v, \sigma')$. Consequently, we can recover $(\tilde{T}, \sigma)$ from $(\tilde{T}, \sigma')$ by inserting the edge $\rho_{\tilde{T}'}v$. If $N^-(\alpha) = \emptyset$, then $(v, x) \in E(\vec{G})$ but $(x, v) \notin E(\vec{G})$ for any $x \in L[t]$. Hence, any informative triple that contains v is induced by X_2 or X_4, and is thus of the form $xy|v$ with $\sigma(x) \neq \sigma(y)$. This implies $v \in \mathsf{child}(\rho_T)$.

On the other hand, if there is a $\beta \in \mathcal{N}$ with $\sigma(\beta) = t$ and $\rho_\beta = \rho_{\tilde{T}}$, we have $(v, u) \in E(\vec{G})$ and $(u, v) \in E(\vec{G})$ with $u \in L[t]$ if and only if $u \in \beta$ by Lemma 4.4(i). Then there is no 3-vertex induced subgraph of $(\vec{G}, \sigma)$ of the form X_1, X_2, X_3, or X_4 that contains both u and v, and any informative triple that contains either u or v is again of the form $xy|v$ and $xy|v$ respectively. As before, this implies $v \in \mathsf{child}(\rho_T)$. Hence, (T, σ) is obtained from (T', σ') by insertion of the edge $\rho_{T'}v$. Since $(\vec{G}', \sigma') = \vec{G}(T', \sigma')$, we conclude that (T, σ) explains $(\vec{G}, \sigma)$, and arrive to the desired contradiction.

Case B: If $|\mathsf{child}(\rho_{\tilde{T}}) \cap L| = 1$, then $(\tilde{T} - v, \sigma')$ is not least resolved since either (a) the root is of degree 1 or (b) there exists no $u \in \mathsf{child}(\rho_{\tilde{T}}) \setminus \{v\}$ such that $\sigma(u) \neq \{s, t\}$ (see Thm. 4.1). In the latter case, the graph $(\vec{G}', \sigma')$ is not connected. To convert $(\tilde{T} - v, \sigma')$ into the least resolved tree $(\tilde{T}', \sigma')$, we need to contract all edges $\rho_{\tilde{T}}u$ with $u \in \mathsf{child}(\rho_{T'}) \setminus \{v\}$. Clearly, we can recover $(\vec{G}, \sigma)$ from $(\vec{G}', \sigma')$ by reverting the prescribed steps. Analogous arguments as in Case A show that again any informative triple in $\mathcal{R}(\vec{G}, \sigma)$ that contains v is of the form $xy|v$ with $\sigma(x) \neq \sigma(y)$. If $(\vec{G}'\sigma')$ is connected, then any triple in $\mathcal{R}(\vec{G}, \sigma) \setminus \mathcal{R}(\vec{G}', \sigma')$ is of this form and hence, as above, we conclude that $v \in \mathsf{child}(\rho_T)$ and $(\vec{G}, \sigma) = \vec{G}(T, \sigma)$. If $(\vec{G}'\sigma')$ is not connected, then $\mathcal{R}(\vec{G}, \sigma) \setminus \mathcal{R}(\vec{G}', \sigma')$ contains also all triples $xy|z$ induced by X_1 and X_3 that emerged from connecting all components of $(\vec{G}', \sigma')$ by insertion of v. However, since $\mathrm{lca}(x, y, z) = \rho_{\tilde{T}}$, we conclude that $v \in \mathsf{child}(\rho_T)$ and thus, $(\vec{G}, \sigma) = \vec{G}(T, \sigma)$ again yields the desired contradiction. □

We finally arrive at the main result of this section.

Theorem 4.6. *A connected properly 2-colored digraph $(\vec{G}, \sigma)$ is a 2-BMG if and only if $(\vec{G}, \sigma) = \vec{G}(\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma)), \sigma)$.*

Proof. If $(\vec{G}, \sigma)$ is a 2-BMG, then Thm. 4.5 guarantees that $(\vec{G}, \sigma) = \vec{G}(\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma)), \sigma)$. If $(\vec{G}, \sigma)$ is not a 2-BMG, then either $\mathcal{R}(\vec{G}, \sigma)$ is inconsistent or its Aho tree $\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma))$ explains a different graph $\vec{G}(T, \sigma) \neq (\vec{G}, \sigma)$ because by assumption $(\vec{G}, \sigma)$ cannot be explained by any tree. □

If $(\vec{G}, \sigma)$ is not connected, then the informative triples of Def. 4.8 are not sufficient by themselves to infer a tree that explains $(\vec{G}, \sigma)$. However, it follows from Thm. 4.1 and 4.6, that the desired tree (T, λ) can be obtained by attaching the Aho trees of the connected components as children of the root of (T, λ). It can be understood as the Aho tree of the triple set

$$\mathcal{R}(\vec{G}, \sigma) = \bigcup_i \mathcal{R}(\vec{G}_i, \sigma_i) \cup \mathcal{R}_C(\vec{G}, \sigma) \tag{6}$$

where the $\mathcal{R}(\vec{G}_i, \sigma_i)$ are the sets of informative triples of the connected components and $\mathcal{R}_C(\vec{G}, \sigma)$ consists of all triples of the form $xy|z$ with $x, y \in L(\vec{G}_i)$ and $z \in L(\vec{G}_j)$ for all pairs $i \neq j$. The triple set $\mathcal{R}_C(\vec{G}, \sigma)$ simply specifies the connected components of $(\vec{G}, \sigma)$. Note that with this augmented definition of $\mathcal{R}$, Thm. 4.6 remains true also for 2-BMGs that are not connected.

In this section we generalize the results about 2-BMGs to an arbitrary number of colors. As in the 2-color case, we write $x\mathsf{R}y$ if and only if x and y have the same in- and out-neighbors. Recall that, for given colors $r, s, t \in S$, we write $(\vec{G}_{st}, \sigma_{st}) := (\vec{G}[L[s] \cup L[t]], \sigma_{|L[s]\cup L[t]})$ and $(\vec{G}_{rst}, \sigma_{rst}) := (\vec{G}[L[r] \cup L[s] \cup L[t]], \sigma_{|L[r]\cup L[s]\cup L[t]})$ for the respective induced subgraphs (see Section 3.2). Since $\vec{G}$ is multipartite and every vertex has at least one out-neighbor of each color except its own, we can conclude also for general BMGs that $x\mathsf{R}y$ implies $\sigma(x) = \sigma(y)$. Denote by $x\mathsf{R}_{st}y$ the thinness relation of Def. 4.3 on $(\vec{G}_{st}, \sigma_{st})$.

Observation 4.4. *If $\sigma(x) = \sigma(y) = s$, then $x\mathsf{R}y$ holds if and only if $x\mathsf{R}_{st}y$ for all colors $t \neq s$.*

We can therefore think of the relation R as the common refinement of the relations R_{st} based on the induced 2-BMGs for all colors s, t. In particular, therefore, all elements within an R-class of an n-BMG appear as sibling leaves in the different least resolved trees, each explaining one of the induced 2-BMGs. Next we generalize the notion of roots.

Definition 4.9. *Let $(\vec{G}, \sigma)$ be an n-BMG explained by some tree (T, σ) and suppose $\sigma(\alpha) = r \neq s$. Then the* root ρ_α of the R-class α w.r.t. color s *is*

$$\rho_{\alpha,s} = \max_{\substack{x\in\alpha \\ y\in N_s^+(\alpha)}} \operatorname{lca}(x, y).$$

Observation 4.5. *Consider an n-BMG $(\vec{G}, \sigma)$ that is explained by a tree (T, σ). By Obs. 4.2, the subgraph $(\vec{G}_{st}, \sigma_{st})$ induced by any two distinct colors $s, t \in S$ is a 2-BMG and thus explained by a corresponding least resolved tree (T_{st}, σ_{st}). Uniqueness of this least resolved tree implies that the tree (T, σ) must display (T_{st}, σ_{st}). In other words, (T, σ) is a refinement of (T_{st}, σ_{st}).*

Observation 4.6. *Let $(\vec{G}, \sigma)$ be an n-BMG that is explained by a tree (T, σ), and $a, b, c \in V(\vec{G})$ leaves of three distinct colors. Then the 3-BMG $\vec{G}(T_{|\{a,b,c\}}, \sigma_{|\{a,b,c\}})$ is the complete graph on $\{a, b, c\}$ with bidirectional edges.*

Therefore no further refinement can be obtained from triples of three different colors. Thus the 2-colored triples inferred from the induced 2-BMGs for all color pairs may already be sufficient to construct (T, σ). This suggests, furthermore, that every n-BMG is explained by a unique least resolved tree. An important tool for addressing this conjecture is the following generalization of Condition (vi) of Lemma 4.3.

Lemma 4.14. *Let $(\vec{G}, \sigma)$ be a (not necessarily connected) n-BMG explained by (T, σ) and let α be an R-class of $(\vec{G}, \sigma)$. Then $N_s^+(\alpha) = L(T(\rho_{\alpha,s})) \cap L[s]$ for all $s \in S \setminus \{\sigma(\alpha)\}$.*

Proof. The definition of $\rho_{\alpha,s}$ implies $N_s^+(\alpha) \subseteq L(T(\rho_{\alpha,s})) \cap L[s]$. In particular, there is a leaf $y \in N_s^+(\alpha)$ such that $\operatorname{lca}(y, \alpha) = \rho_{\alpha,s}$. Now consider an arbitrary leaf $x \in L(T(\rho_{\alpha,s})) \cap L[s]$. By construction, we have $\operatorname{lca}(x, \alpha) \preceq \rho_{\alpha,s} = \operatorname{lca}(y, \alpha)$ and therefore $x \in N_s^+(\alpha)$, which completes the proof. □

We are now in the position to characterize the redundant edges.

Lemma 4.15. *Let $(\vec{G},\sigma)$ be a (not necessarily connected) n-BMG explained by (T,σ). Then the inner edge $e=uv$ is redundant in (T,σ) if and only if for every color $s\in\sigma(L(T(u))\setminus L(T(v)))$, there is no* R*-class $\alpha\in\mathcal{N}$ with $v=\rho_{\alpha,s}$.*

Proof. Let (T_e,σ) be the tree that is obtained from (T,σ) by contraction of the edge $e=uv$ and assume that (T_e,σ) explains $(\vec{G},\sigma)$. Since (T,σ) is phylogenetic, there exists a leaf $y\in L(T(u))\setminus L(T(v))$ of some color $s\in\sigma(L(T(u))\setminus L(T(v)))$. Assume that there is an R-class α of $(\vec{G},\sigma)$ such that $v=\rho_{\alpha,s}$. Note that $s\neq\sigma(\alpha)$ by definition of $\rho_{\alpha,s}$. Lemma 4.14 implies that $y\notin N^+(\alpha)$ in $(\vec{G},\sigma)$. After contraction of e, we have $\mathrm{lca}(\alpha,y)=\rho_{\alpha,s}$, thus $y\in N^+(\alpha)$ by Lemma 4.14. Hence, (T_e,σ) does not explain $(\vec{G},\sigma)$; a contradiction.

Conversely, assume that for every $s\in\sigma(L(T(u))\setminus L(T(v)))$, there is no $\alpha\in\mathcal{N}$ such that $v=\rho_{\alpha,s}$, i.e., for every $\alpha\in\mathcal{N}$ and every color $s\neq\sigma(\alpha)$ we either have (i) $v\succ\rho_{\alpha,s}$, (ii) $v\prec\rho_{\alpha,s}$, or (iii) v and $\rho_{\alpha,s}$ are incomparable. In the first two cases, contraction of e implies $v\succeq\rho_{\alpha,s}$ or $v\preceq\rho_{\alpha,s}$ in (T_e,σ), respectively. Therefore, and since $L(T(w))=L(T_e(w))$ for any w incomparable to v, we have $L(T(w))=L(T_e(w))$ for any node $w\neq v$. Moreover, it follows from Lemma 4.14 that $N_s^+(\alpha)=\{y\mid y\in L(T(\rho_{\alpha,s})),\sigma(y)=s\}$. This implies that the set $N_s^+(\alpha)$ remains unchanged after contraction of e for all R-classes α and all color $s\in S$. In other words, the in- and out-neighborhood of any leaf remain the same in (T_e,σ). Hence, we conclude that (T,σ) and (T_e,σ) explain the same graph $(\vec{G},\sigma)$. □

Before we consider the general case, we show that 3-BMGs like 2-BMGs are explained by unique least resolved trees.

Lemma 4.16. *Let $(\vec{G},\sigma)$ be a connected 3-BMG. Then there exists a unique least resolved tree (T,σ) that explains $(\vec{G},\sigma)$.*

Proof. This proof uses arguments very similar to those in the proof of the uniqueness result for 2-BMGs. In particular, as in the proof of Thm. 4.2, we assume for contradiction that there exist 3-colored digraphs that are explained by two distinct least resolved trees. Let $(\vec{G},\sigma)$ be a minimal graph (w.r.t. the number of vertices) that is explained by the two distinct least resolved trees (T_1,σ) and (T_2,σ). W.l.o.g. we can choose a vertex v and assume that its color is $r\in S$, i.e., $v\in L[r]$. Using the same notation as in the proof of Thm. 4.2, we write (T_1',σ') and (T_2',σ') for the trees that are obtained by deleting v from (T,σ). These trees explain the uniquely defined graphs $(\vec{G}_1',\sigma')$ and $(\vec{G}_2',\sigma')$, respectively. Again, Lemma 4.1 implies that $(\vec{G}',\sigma'):=(\vec{G}[L\setminus\{v\}],\sigma')$ is a subgraph of both $(\vec{G}_1',\sigma')$ and $(\vec{G}_2',\sigma')$. Similar to the case of 2-BMGs, we characterize the additional edges that are inserted into $(\vec{G}_1',\sigma')$ and $(\vec{G}_2',\sigma')$ compared to $(\vec{G}',\sigma')$ in order to show that $(\vec{G}_1',\sigma')=(\vec{G}_2',\sigma')$. Assume that (u,y) is an edge in $(\vec{G}_1',\sigma')$ but not in $(\vec{G}',\sigma')$. By analogous arguments as in the proof of Thm. 4.2, we find that $(u,v)\in E(\vec{G})$ and in particular $N_r^+(u)=\{v\}$, i.e., u has no out-neighbors of color r in $(\vec{G}',\sigma')$.

Moreover, we have $u \in L[s]$, where $s \in S \setminus \{r\}$. Similar to the 2-color case, we now determine the outgoing arcs of u in $(\vec{G}'_1, \sigma')$ and $(\vec{G}'_2, \sigma')$ by reconstructing the local structure of (T_1, σ) and (T_2, σ) in the vicinity of v.

Obs. 4.2 implies that the least resolved tree (T_{rs}, σ_{rs}) explaining $(\vec{G}_{rs}, \sigma_{rs})$ is displayed by both (T_1, σ) and (T_2, σ). The local structure of (T_{rs}, σ_{rs}) around v is depicted in Fig. 9. Using the notation in the figure, $\{v\}$ is an R-class by itself, $\alpha = \{v\}$, there is an R-class $\beta' \subseteq L[s]$ with $N_r^+(\beta') = \{\alpha\}$ and $N_s^+(\alpha) = \{\beta'\}$, and there may or may not exist an R-class $\beta \subseteq L[s]$ with $N_r^+(\beta) = N_r^+(\beta') = \{\alpha\}$ and $N_s^+(\alpha) \cap \beta' = \emptyset$. In addition, we have $\gamma \subseteq L[r]$, which is the $\preceq$-minimal R-class of color r such that $\rho_\gamma \succ \rho_\beta, \rho_{\beta'}$. Recall that (u, c) with $c \in \gamma$ are all the edges on $L[r] \times L[s]$ that have been additionally inserted in both $(\vec{G}'_1, \sigma')$ and $(\vec{G}'_2, \sigma')$. Since every R-class has at least one out-neighbor of each color and given the relationship between α and β', there exists an R-class $\delta \subseteq L[t]$, where $t \in S \setminus \{r, s\}$, with $\alpha \subseteq N_r^+(\delta)$ and $\beta' \subseteq N_s^+(\delta)$ such that there is no other $\delta' \subseteq L[t]$ with $\rho_{\delta'} \prec \rho_\delta$. If $N_r^+(\delta) \setminus \{\alpha\} \neq \emptyset$, then $\rho_\delta \succeq \rho_\gamma$ by Lemma 4.14, and in particular there is no additional edge of the form (w, a) with $w \in L[t]$ and $a \in L[r]$ that is contained in $(\vec{G}'_1, \sigma')$ and/or $(\vec{G}'_2, \sigma')$ but not in $(\vec{G}', \sigma')$. Therefore only edges of the form (u, c) with $c \in \gamma$ are additionally inserted into $(\vec{G}'_1, \sigma')$ and $(\vec{G}'_2, \sigma')$, and we conclude that $(\vec{G}'_1, \sigma') = (\vec{G}'_2, \sigma')$, which implies $(T'_1, \sigma') = (T'_2, \sigma')$ and therefore, since v was arbitrary, $(T_1, \sigma') = (T_2, \sigma')$; a contradiction.

Now consider the case $N_r^+(\delta) \setminus \{\alpha\} = \emptyset$. Since $\gamma \notin N_r^+(\delta)$, Lemma 4.14 ensures that $\rho_\delta \not\succeq \rho_\gamma$. The roots ρ_γ and ρ_δ are comparable since α is an out-neighbor of both γ and δ. Thus $\rho_\delta \prec \rho_\gamma$ and hence, $N_r^+(\delta) = \{\gamma\}$ in (T'_1, σ') as well as in (T'_2, σ') after deletion of v. We still need to distinguish two cases: either we have $N_s^+(\delta) = \{\beta'\}$ or $N_s^+(\delta) = \{\beta', \beta\}$. In the first case, we have $\rho_\delta = \rho_{\beta'} = \rho_\alpha$ in (T'_1, σ') as well as in (T'_2, σ'). In the second case, we obtain $\rho_\delta = \rho_\beta$, again this holds for both (T'_1, σ') and (T'_2, σ'). As before, we can conclude that $(T'_1, \sigma') = (T'_2, \sigma')$ and therefore $(T_1, \sigma') = (T_2, \sigma')$; a contradiction. □

If a 3-BMG $(\vec{G}, \sigma)$ is not connected, we can build a least resolved tree (T, σ) analogously to the case of 2-BMGs: we first construct the unique least resolved tree (T_i, σ_i) for each component $(\vec{G}_i, \sigma_i)$. Using Thm. 4.1 we then insert an additional root for (T, σ) to which the roots of the $(\vec{G}_i, \sigma_i)$ are attached as children. We proceed by showing that this construction corresponds to the unique least resolved tree.

Theorem 4.7. *Let $(\vec{G}, \sigma)$ be a (not necessarily connected) n-BMG with $n \in \{2, 3\}$. Then there exists a unique least resolved tree (T, σ) that explains $(\vec{G}, \sigma)$.*

Proof. Denote by $(\vec{G}_i, \sigma_i)$ the connected components of $(\vec{G}, \sigma)$. By Thm. 4.2 and Lemma 4.16 there is a unique least resolved tree (T_i, σ_i) that explains $(\vec{G}_i, \sigma_i)$. Hence, if $(\vec{G}, \sigma)$ is connected, we are done.

Now assume that there are at least two connected components. Let (T, σ) be a least resolved tree that explains $(\vec{G}, \sigma)$. Thm. 4.1 implies that there is a vertex $u \in \mathsf{child}(\rho_T)$ such that $L(\vec{G}_i) \subseteq L(T(u))$ for each connected component $(\vec{G}_i, \sigma_i)$. Hence, the subtree $(T(u), \sigma_{|L(T(u))})$ displays the least resolved tree

(T_i, σ_i) explaining $(\vec{G}_i, \sigma_i)$. Moreover, since (T, σ) is least resolved, $\rho_T u$ is a relevant edge, i.e., there must be a color $s \in \sigma(L(T) \setminus L(T(u)))$ and an R-class α such that $u = \rho_{\alpha,s}$ by Lemma 4.15.

This implies in particular that there exists a leaf $x \in L(T(u)) \cap L[s]$. Lemma 4.14 now implies that the elements of α are connected to any element of color s in the subtree $(T(u), \sigma_{|L(T(u))})$. Furthermore, any leaf $y \in L(T(u))$ of color different from s has at least one out-neighbor of color s in $L(T(u))$. Hence, we can conclude that the graph $\vec{G}(T(u), \sigma_{|L(T(u))})$ induced by the subtree $(T(u), \sigma_{|L(T(u))})$ is connected.

Since $L(\vec{G}_i) \subseteq L(T(u))$ and $(T(u), \sigma_{|L(T(u))})$ explains the *maximal connected* subgraph $(\vec{G}_i, \sigma_i)$, we conclude that $\vec{G}(T(u), \sigma_{|L(T(u))}) = (\vec{G}_i, \sigma_i)$. By construction, both $(T(u), \sigma_| L(T(u)))$ and (T_i, σ_i) are least resolved trees explaining the same graph, hence Thm. 4.2 and Lemma 4.16 imply $(T(u), \sigma_{|L(T(u))}) = (T_i, \sigma_i)$. In particular, thus, $\rho_{T_i} = u$.

As a consequence, any least resolved tree (T, σ) that explains $(\vec{G}, \sigma)$ must be composed of the disjoint trees (T_i, σ_i) that are linked to the root via the relevant edge $\rho_T \rho_{T_i}$. Since every (T_i, σ_i) and the construction of the edges $\rho_T \rho_{T_i}$ is unique, (T, σ) is unique. □

The characterization of redundant edges in trees explaining 2-BMGs together with the uniqueness of the least resolved trees for 2-BMGs and 3-BMGs can be used to characterize redundant edges in the general case, thereby establishing the existence of a unique least resolved tree for n-BMGs.

Theorem 4.8. *For any connected n-BMG $(\vec{G}, \sigma)$, there exists a unique least resolved tree (T', σ) that explains $(\vec{G}, \sigma)$. The tree (T', σ) is obtained by contraction of all redundant edges in an arbitrary tree (T, σ) with leaf set L that explains $(\vec{G}, \sigma)$. The set of all redundant edges in (T, σ) is given by*

$$\mathfrak{E}_T = \{e = uv \mid v \notin L, v \neq \rho_{\alpha,s} \text{ for all } s \in \sigma(L(T(u)) \setminus L(T(v))) \text{ and } \alpha \in \mathcal{N}\}.$$

Moreover, (T', σ) is displayed by (T, σ).

Proof. Using arguments analogous to the 2-color case one shows that there is a least resolved tree (T', σ) that can be obtained from (T, σ) by contraction of all redundant edges. The set of redundant edges is given by $\mathfrak{E}_T$ (cf. Lemma 4.15). By construction, (T', σ) is displayed by (T, σ). It remains to show that (T', σ) is unique. Obs. 4.2 implies that for any pair of distinct colors s and t the corresponding unique least resolved tree (T_{st}, σ_{st}) is displayed by (T', σ). The same is true for the least resolved tree (T_{rst}, σ_{rst}) for any three distinct colors $r, s, t \in S$. Since for any 2-BMG as well as for any 3-BMG, the corresponding least resolved tree is unique (see Thm. 4.2 and Lemma 4.16), it follows for any three distinct leaves $x, y, z \in L[r] \cup L[s] \cup L[t]$ that there is either a unique triple that is displayed by (T_{rst}, σ_{rst}) or the least resolved tree (T_{rst}, σ_{rst}) contains no triple on x, y, z. Note that we do not require that the colors r, s, t are pairwise distinct. Instead, we use the notation (T_{rst}, σ_{rst}) to also include the trees explaining the induced 2-BMGs. Obs. 4.2 then implies that $\mathcal{R}^* := \bigcup_{r,s,t \in S} r(T_{rst}) \subseteq r(T')$. Now assume that there are two distinct least resolved trees (T_1, σ) and (T_2, σ) that explain $(\vec{G}, \sigma)$. In the following

we show that any triple displayed by T_1 must be displayed by T_2 and thus, $r(T_1) = r(T_2)$.

Fig. 14 shows that there may be triples $xy|z \in r(T_1) \setminus \mathcal{R}^*$. Assume, for contradiction, that $xy|z \notin r(T_2) \setminus \mathcal{R}^*$. Fix the notation such that $z \in \alpha$, $\sigma(x) = r$, $\sigma(y) = s$, and $\sigma(z) = t$. We do not assume here that the colors r, s, t are necessarily pairwise distinct.

In the remainder of the proof, we will make frequent use of the following:

Observation: *If the tree T is a refinement of T', then we have $u \preceq_{T'} v$ if and only if $u \preceq_T v$ for all $u, v \in V(T')$.*

In particular, $u \prec_{T'} v$ implies $u \prec_T v$. The converse of the latter statement is still true if u is a leaf in T' but not necessarily for arbitrary inner vertices u and v.

Let $u = \mathrm{lca}_{T_1}(x, y, z)$. The assumption $xy|z \in r(T_1)$ implies that there is a vertex $v \in \mathsf{child}_{T_1}(u)$ such that $v \succeq \mathrm{lca}_{T_1}(x, y)$. Since (T_1, σ) is least resolved, the characterization of relevant edges ensures that there is a color $p \in \sigma(L(T_1(u)) \setminus L(T_1(v)))$ and an R-class β with $\sigma(\beta) = q$ such that $v = \rho_{\beta,p}$. In particular, there must be leaves $a \in L(T_1(v))$ and $a^* \in L(T_1(u)) \setminus L(T_1(v))$ with $\sigma(a) = \sigma(a^*) = p$. As a consequence we know that $a^* \notin N_p^+(b)$ for any $b \in \beta$.

We continue to show that the edge uv must also be contained in the least resolved tree (T_{pq}, σ_{pq}) that explains the (not necessarily connected) graph $(\vec{G}_{pq}, \sigma_{pq})$. By Thm. 4.7, (T_{pq}, σ_{pq}) is unique. Assume, for contradiction, that uv is not an edge in T_{pq}. Recalling the arguments in Obs. 4.5, the tree (T_1, σ) must display (T_{pq}, σ_{pq}). Thus, if uv is not an edge in T_{pq}, then $v^* := u = v$ in T_{pq}. By construction, we therefore have $v^* = \rho_{\beta,p}$ in (T_{pq}, σ_{pq}). Since (T_{pq}, σ_{pq}) is least resolved, it follows from Cor. 4.3 that $b \in \mathsf{child}(v^*)$ for all $b \in \beta$ in (T_{pq}, σ_{pq}). The latter, together with $a, a^* \preceq_{T_{pq}} v^*$, implies that $\mathrm{lca}_{T_{pq}}(a, \beta) = \mathrm{lca}_{T_{pq}}(a^*, \beta) = v^*$. However, this implies $a^* \in N_p^+(\beta)$; a contradiction.

To summarize, the edge uv must be contained in the least resolved tree (T_{pq}, σ_{pq}). Moreover, by Obs. 4.5, (T_{pqo}, σ_{pqo}) is a refinement of (T_{pq}, σ_{pq}) for every color $o \in S$. Hence, we have $v \prec_{T_{pqo}} u$, which is in particular true for the color $o \in \{r, s, t\}$. Moreover, we know that $x \prec_{T_{pqr}} v$ and $y \prec_{T_{pqs}} v$ because (T_1, σ) is a refinement of both (T_{pqr}, σ_{pqr}) and (T_{pqs}, σ_{pqs}).

Since (T_2, σ) is also a refinement of both (T_{pqr}, σ_{pqr}) and (T_{pqs}, σ_{pqs}), we have $x, y \prec_{T_2} v \prec_{T_2} u$. Furthermore, $v \prec_{T_1} \mathrm{lca}_{T_1}(v, z) = u$ and $z \not\preceq_{T_1}$ implies that $z \prec_{T_{pqt}} u$ and $z \not\preceq_{T_{pqt}} v$. Therefore, $z \prec_{T_2} u$ and $z \not\preceq_{T_2} v$. Combining these facts about partial order of the vertices v, u, x, y, and z in T_2, we obtain $xy|z \in r(T_2)$; a contradiction.

Hence, $r(T_1) = r(T_2)$. Since $r(T_1)$ uniquely identifies the structure of T_1 (cf. Semple and Steel [202, Thm. 6.4.1]), we conclude that $(T_1, \sigma) = (T_2, \sigma)$. The least resolved tree explaining $(\vec{G}, \sigma)$ is therefore unique.

□

Corollary 4.4. *Every n-BMG $(\vec{G}, \sigma)$ is explained by the unique least resolved tree (T, σ) consisting of the least resolved trees (T_i, G_i) explaining the connected*

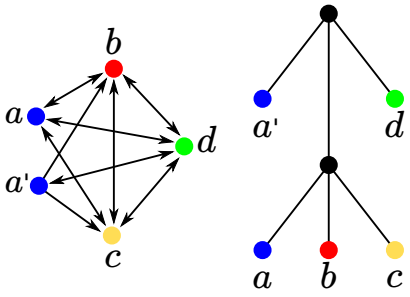


Fig. 14. A connected graph $(\vec{G}, \sigma)$ and the corresponding least resolved tree (T, σ) on five vertices of four colors: blue (a and a'), red (b), yellow (c), and green (d). The triple $bc|d$ is displayed by (T, σ) but it is not displayed by the least resolved tree (T', σ') that explains the induced subgraph $(\vec{G}', \sigma')$ with $V(\vec{G}') = \{b, c, d\}$ since (T', σ') is simply the star tree on $\{b, c, d\}$. Hence, $bc|d \notin \mathcal{R}^* = \bigcup_{r,s,t \in S} r(T_{rst})$.

components $(\vec{G}_i, \sigma_i)$ *and an additional root* ρ_T *to which the roots of the* (T_i, G_i) *are attached as children.*

Proof. It is clear from the construction that (T, σ) explains $(\vec{G}, \sigma)$. The proof that this is the only least resolved tree parallels the arguments in the proof of Thm. 4.7 for 2-BMGs and 3-BMGs. □

As a tree is determined by all its triples, it is now clear that the construction of a tree that explains a connected n-BMG is essentially a supertree problem: it suffices to find a tree, if it exists, that displays the least resolved trees explaining the induced subgraphs on 3 colors. In the following, we write

$$R := \bigcup_{s,t \in S} r(T^*_{st}) \tag{7}$$

for the union of all triples in the least resolved trees (T^*_{st}, σ_{st}) explaining the 2-colored subgraphs $(\vec{G}_{st}, \sigma_{st})$ of $(\vec{G}, \sigma)$. In contrast, the set of all *informative* triples of $(\vec{G}, \sigma)$, as specified in Def. 4.8, is denoted by $\mathcal{R}(\vec{G}, \sigma)$. As an immediate consequence of Lemma 4.12 we have

$$\mathcal{R}(\vec{G}, \sigma) \subseteq R. \tag{8}$$

Theorem 4.9. *A connected properly* n*-colored digraph* $(\vec{G}, \sigma)$ *is an* n*-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,* $(\mathrm{Aho}(R), \sigma)$ *is the unique least resolved tree that explains* $(\vec{G}, \sigma)$*.*

Proof. Let $(\vec{G}, \sigma)$ be an n-BMG that is explained by a tree (T, σ). Moreover, let s and t be two distinct colors of $(\vec{G}, \sigma)$ and $L' := L[s] \cup L[t]$ the subset of vertices with color s and t, respectively. Obs. 4.2 states that the induced subgraph $(\vec{G}[L'], \sigma_{|L'})$ is a 2-BMG that is explained by $(T_{|L'}, \sigma_{|L'})$. In particular, the least resolved tree $(T^*_{|L'}, \sigma_{|L'})$ of $(T_{|L'}, \sigma_{|L'})$ also explains $(\vec{G}[L'], \sigma_{|L'})$ and $T^*_{|L'} \leq T_{|L'} \leq T$ by Thm. 4.8, i.e., $r(T^*_{|L'}) \subseteq r(T)$. Since this holds for all pairs of two distinct colors, the union of the triples obtained from the set of all least resolved 2-BMG trees R is displayed by T. In particular, therefore, R is consistent.

Conversely, suppose that $(\vec{G}[L'], \sigma_{|L'})$ is a 2-BMG for any two distinct colors s, t and R is consistent. Let $(\mathrm{Aho}(R), \sigma)$ be the tree that is constructed by `BUILD` for the input set R with leaf coloring as in $(\vec{G}, \sigma)$. This tree displays R and is a least resolved tree [3] in the sense that we cannot contract any edge in $\mathrm{Aho}(R)$ without loosing a triple from R. By construction, any triple that is displayed by (T_{st}, σ_{st}) is also displayed by $(\mathrm{Aho}(R), \sigma)$, i.e. $T_{st} \leq \mathrm{Aho}(R)$. Hence, for any $\alpha \in \mathcal{N}$ and any color $s \neq \sigma(\alpha)$, the out-neighborhood $N_s^+(\alpha)$ is the same w.r.t. (T_{st}, σ_{st}) and w.r.t. $(\mathrm{Aho}(R), \sigma)$. Since this is true for any R-class of $(\vec{G}, \sigma)$, also all in-neighborhoods are the same in $(\mathrm{Aho}(R), \sigma)$ and the corresponding (T_{st}, σ_{st}). Therefore we conclude that $(\mathrm{Aho}(R), \sigma)$ explains $(\vec{G}, \sigma)$, i.e., $(\vec{G}, \sigma)$ is an n-BMG.

In order to see that $(\mathrm{Aho}(R), \sigma)$ is a least resolved tree explaining $(\vec{G}, \sigma)$, we recall that the contraction of an edge leaves at least one triple unexplained, see Semple [201, Prop. 4.1]. Since R consists of all the triples $r(T_{st})$ that in turn uniquely identify the structure of (T_{st}, σ_{st}) (cf. Semple and Steel [202, Thm. 6.4.1]), none of these triples is dispensable. The contraction of an edge in $\mathrm{Aho}(R)$ therefore yields a tree that no longer displays (T_{st}, σ_{st}) for some pair of colors s, t and thus, no longer explains $(\vec{G}, \sigma)$. Thus $(\mathrm{Aho}(R), \sigma)$ contains no redundant edges and we can apply Thm. 4.8 to conclude that $(\mathrm{Aho}(R), \sigma)$ is the unique least resolved tree that explains $(\vec{G}, \sigma)$. □

Fig. 15 summarizes the construction of the least resolved tree from the 3-colored digraph $(\vec{G}, \sigma)$ shown in Fig. 15(B). For simplicity we assume that we already know that $(\vec{G}, \sigma)$ is indeed a 3-BMG. For each of the three colors the example has four genes. In addition to singletons there are three non-trivial R-classes $\alpha = \{a_2, a_3, a_4\}$, $\beta = \{b_3, b_4\}$ and $\gamma = \{c_3, c_4\}$. Following Thm. 4.9, we extract for each of the three pairs of colors the induced subgraphs $(\vec{G}_{st}, \sigma_{st})$ and construct the least resolved trees that explain them (Fig. 15(C)). Extracting all triples from these least resolved trees on two colors yields the triple set $\mathcal{R}(\vec{G}, \sigma)$, which in this case is consistent. Thm. 4.9 implies that the tree $(\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma)), \sigma)$ (shown in the lower right corner) explains $(\vec{G}, \sigma)$ and is in particular the unique least resolved tree w.r.t. $(\vec{G}, \sigma)$.

We close this section by showing that in fact the informative triples of all $(\vec{G}_{st}, \sigma_{st})$ are already sufficient to decide whether $(\vec{G}, \sigma)$ is an n-BMG or not. More precisely, we show

Lemma 4.17. *If $(\vec{G}, \sigma)$ is an n-BMG, then* $\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma)) = \mathrm{Aho}(R)$.

Proof. We first observe that the two triple sets R and $\mathcal{R} := \mathcal{R}(\vec{G}, \sigma)$ have the same Aho tree $\mathrm{Aho}(R) = \mathrm{Aho}(\mathcal{R})$ if, in each step of `BUILD`, the respective Aho graphs $[R, L']$ and $[\mathcal{R}, L']$, as defined in Chapter 3, have the same connected components. It is not necessary, however, that $[R, L']$ and $[\mathcal{R}, L']$ are isomorphic. In the following we set $T = \mathrm{Aho}(R)$.

If T is the star tree on L, then $\mathcal{R} \subseteq R = \emptyset$, thus $[R, L] = [\mathcal{R}, L]$ is the edgeless graph on L. Hence, in particular, $\mathrm{Aho}(\mathcal{R}) = \mathrm{Aho}(R)$.

Now suppose T is not the star tree. Then there is a vertex $w \in V^0(T)$ such that $L(T(w)) = \mathsf{child}(w)$. For simplicity, we write $L_w := L(T(w))$. Since $(T(w), \sigma_{|L_w})$ is a star tree, we can apply the same argument again to conclude that $[R_{|L_w}, L_w] = [\mathcal{R}_{|L_w}, L_w]$, hence both Aho graphs have the same connected

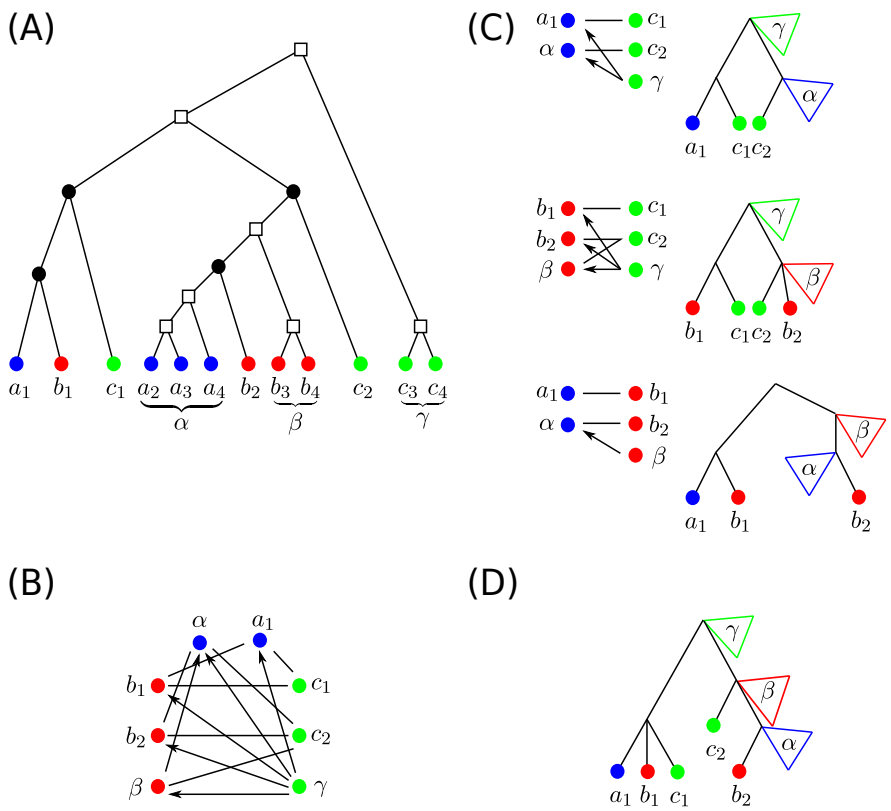


Fig. 15. Construction of the least resolved tree explaining the colored best match graph. Panel (A) recalls the event-labeled gene tree of the evolutionary scenario shown in Fig. 5. There are three R-classes with more than one element: $\alpha = \{a_2, a_3, a_4\}$, $\beta = \{b_3, b_4\}$ and $\gamma = \{c_3, c_4\}$ in the 3-BMG graph $(\vec{G}, \sigma)$, which is shown in Panel (B). For simplicity of presentation, the R-classes are already collapsed into single vertices and bidirectional edges are represented as solid lines without arrow heads. Panel (C) lists the three induced subgraphs of $(\vec{G}, \sigma)$ on two colors together with their least resolved trees. By construction, $(\vec{G}, \sigma)$ is the union of the three subgraphs on two colors. (D) The leaf-labeled Aho tree for the set of all triples obtained from the least resolved trees shown in (C). This tree explains the graph $(\vec{G}, \sigma)$ and is the unique least resolved tree w.r.t. $(\vec{G}, \sigma)$.

components. Now let $u = \rho_T$ and assume by induction that $[R_{|L_{u'}}, L_{u'}]$ and $[\mathcal{R}_{|L_{u'}}, L_{u'}]$ have the same connected components for every $u' \prec_T u$ and thus, in particular, for $v \in \mathsf{child}(u)$. Consequently, for any $v_i \in \mathsf{child}(v)$ the set L_{v_i} is connected in $[\mathcal{R}_{|L_v}, L_v]$. Since $\mathcal{R}_{|L_v} \subseteq \mathcal{R}_{|L_u}$, the set L_{v_i} must also be connected in $[\mathcal{R}_{|L_u}, L_u]$ for every $v_i \in \mathsf{child}(v)$ (cf. Prop. 8 in [26]). It remains to show that all L_{v_i} are connected in $[\mathcal{R}_{|L_u}, L_u]$.

Since (T, σ) is least resolved w.r.t. $(\vec{G}, \sigma)$, it follows from Thm. 4.8 that $v = \rho_{\alpha,s}$ for some color $s \in \sigma(L(T(u)) \setminus L(T(v)))$ and an R-class α with $\sigma(\alpha) \neq s$. In particular, therefore, $s \notin \sigma(L_{v_i})$ if $\alpha \in L_{v_i}$ (say $i = 1$). By definition of s, there must be a $v_j \in \mathsf{child}(v) \setminus \{v_1\}$ (say $j = 2$) such that $s \in \sigma(L_{v_2})$. Let $y \in L_{v_2} \cap L[s]$. Lemma 4.14 implies $y \in N_s^+(\alpha)$, i.e., $(\alpha, y) \in E(\vec{G})$. Moreover, by definition of s, there must be a leaf $y' \in L(T(u)) \setminus L(T(v))$ of color s. Since $\mathrm{lca}(\alpha, y) \prec_T \mathrm{lca}(\alpha, y')$, we have $(\alpha, y') \notin E(\vec{G})$, whereas (y', α) may or may

not be contained in $(\vec{G}, \sigma)$. Therefore, the induced subgraph on $\{\alpha, y, y'\}$ is of the form X_1, X_2, X_3, or X_4 and thus provides the informative triple $\alpha y|y'$. It follows that L_{v_1} and L_{v_2} are connected in $[\mathcal{R}_{|L_u}, L_u]$. In particular, this implies that any L_{v_j} with $\sigma(L_{v_j}) \subseteq \sigma(L_v)$ containing s is connected to any L_{v_i} that does not contain s. Since $(\vec{G}, \sigma)$ is connected, such a set L_{v_i} always exists by Thm. 4.1. Now let $L_1 := \{L_{v_j} \mid v_j \in \mathsf{child}(v), s \in \sigma(L_{v_j})\}$ and $L_2 := \{L_{v_i} \mid v_i \in \mathsf{child}(v), s \notin \sigma(L_{v_i})\}$. It then follows from the arguments above that L_1 and L_2 form a complete bipartite graph, hence $[\mathcal{R}_{|L_u}, L_u]$ is connected.

□

As an immediate consequence, Thm. 4.9 can be rephrased as:

Corollary 4.5. *A connected properly n-colored digraph $(\vec{G}, \sigma)$ is an n-BMG if and only if (i) all induced subgraphs $(\vec{G}_{st}, \sigma_{st})$ on two colors are 2-BMGs and (ii) the union $\mathcal{R}$ of informative triples $\mathcal{R}(\vec{G}_{st}, \sigma_{st})$ obtained from the induced subgraphs $(\vec{G}_{st}, \sigma_{st})$ forms a consistent set. In particular, $(\mathrm{Aho}(\mathcal{R}), \sigma)$ is the unique least resolved tree that explains $(\vec{G}, \sigma)$.*

4.5 ALGORITHMIC CONSIDERATIONS

The material in the previous two sections can be translated into practical algorithms that decide for a given colored graph $(\vec{G}, \sigma)$ whether it is an n-BMG and, if this is the case, compute the unique least resolved tree that explains $(\vec{G}, \sigma)$. The correctness of Algorithm 1 follows directly from Thm. 4.9 (for a single connected component) and Thm. 4.1 regarding the composition of connected components. It depends on the construction of the unique least resolved tree for the connected components of the induced 2-BMGs, called `LRTfrom2BMG`() in the pseudocode of Algorithm 1. There are two distinct ways of computing these trees: either by constructing the hierarchy $T(\mathcal{H})$ from the extended reachable sets R' (Algorithm 2) or via constructing the Aho tree from the set of informative triples (Algorithm 3). While the latter approach seems simpler, we shall see below that it is in general slightly less efficient. Furthermore, we use a function `BuildST`() to construct the supertree from a collection of input trees. Together with the computation of Aho() from a set of triples, it will be briefly discussed later in this section.

Let us now turn to analyzing the computational complexity of Algorithms 1, 2, and 3. We start with the building blocks necessary to process the 2-BMG $(\vec{G} = (L, \vec{E}), \sigma)$ and consider performance bounds on individual tasks.

FROM (T, σ) TO $(\vec{G}, \sigma)$. Given a leaf-labeled tree (T, σ) with leaf set L we first consider the construction of the corresponding BMG. The necessary lowest common ancestor queries can be answered in constant time after linear time preprocessing, see e.g. [89, 198]. The lca() function can also be used to express the partial orders among vertices since we have $x \preceq y$ if and only if $\mathrm{lca}(x, y) = y$. In particular, therefore, $\mathrm{lca}(x, y) \preceq \mathrm{lca}(x, y')$ is true if and only if $\mathrm{lca}(\mathrm{lca}(x, y), \mathrm{lca}(x, y')) = \mathrm{lca}(\mathrm{lca}(x, y), y') = \mathrm{lca}(x, y')$. Thus $(\vec{G}, \sigma)$ can be constructed from (T, σ) by computing $\mathrm{lca}(x, y)$ in constant time for each

Algorithm 1 Unique least resolved tree of n-BMG

Require: Vertex-colored digraph $(\vec{G} = (L, \vec{E}), \sigma)$.
  if there is $(x, y) \in \vec{E}$ with $\sigma(x) = \sigma(y)$ **then**
    exit(*"not a BMG"*)
  determine connected components $(\vec{G}_i = (L_i, \vec{E}_i), \sigma_i)$
  if $\sigma(L_i) \neq \sigma(L_j)$ for some components i, j **then**
    exit(*"not a BMG"*)
  for all connected components $(\vec{G}_i, \sigma_i)$ **do**
    for all colors $s, t \in S$, $s \neq t$ **do**
      determine the induced subgraph $(\vec{G}_{st} = (L_{st}, \vec{E}_{st}), \sigma_{st})$ with colors s, t
      determine connected components $(\vec{G}_{st,i}, \sigma_{st,i})$
      for all connected components $(\vec{G}_{st,i}, \sigma_{st,i})$ **do**
        $(T_{st,i}, \sigma_{st,i}) \leftarrow$ `LRTfrom2BMG`$(G_{st,i}, \sigma_{st,i})$
        if $(T_{st,i}, \sigma_{st,i}) = \varnothing$ **then**
          exit(*"not a BMG"*)
      $(T_{st}, \sigma_{st}) \leftarrow$ root r_{st} with children $(T_{st,i}, \sigma_{st,i})$
    $(T_i, \sigma_i) \leftarrow$ `BuildST`$(\bigcup_{s,t}(T_{st}, \sigma_{st}))$
    if $(T_i, \sigma_i) = \varnothing$ **then**
      exit(*"not a BMG"*)
  $(T, \sigma) \leftarrow$ root r with children (T_i, σ_i)
  return (T, σ)

Algorithm 2 Unique least resolved tree of connected 2-BMG

Require: Two-colored connected bipartite digraph $(\vec{G}(L, \vec{E}), \sigma)$.
  compute R-classes
  compute $N^+(\alpha)$ and $N^+(N^+(\alpha))$ for all α
  if (N2) does not hold for all α **then**
    return $\varnothing$
  if (N3) does not hold for all α, β **then**
    return $\varnothing$
  compute table $Y_{\alpha\beta} = 1$ if and only if $N^+(\alpha) \cap N^+(N^+(\beta)) \neq \emptyset$
  if (N1) does not hold for all α, β **then**
    return $\varnothing$
  compute $R(\alpha)$, $Q(\alpha)$, and $R'(\alpha) = R(\alpha) \cup Q(\alpha)$ for all α
  tabulate $P_{\alpha,\beta} = 1$ if and only if $R'(\alpha) \subseteq R'(\beta)$.
  compute Hasse $T(\mathfrak{H})$ diagram by transitive reduction
  if $T(\mathfrak{H})$ is not a tree **then**
    return $\varnothing$
  if there are siblings $R'(\alpha)$ and $R'(\beta)$ in $T(\mathfrak{H})$ with non-empty intersection **then**
    return $\varnothing$
  construct $T^*(\mathfrak{H})$ by attaching the leaves to $T(\mathfrak{H})$
  return $T^*(\mathfrak{H})$

Algorithm 3 Unique least resolved tree of connected 2-BMG via triples

Require: Two-colored connected bipartite digraph $(\vec{G} = (L, \vec{E}), \sigma)$.
  extract informative triple set $\mathcal{R}$ from $(\vec{G}, \sigma)$
  $(T, \sigma) \leftarrow (\mathrm{Aho}(\mathcal{R}), \sigma)$
  compute $\vec{G}(T, \sigma)$
  if $\vec{G}(T, \sigma) = (\vec{G}, \sigma)$ **then**
    return (T, σ)
  else
    return $\varnothing$

leaf x and each $y \in L[s]$. Since the last common ancestors for fixed x are comparable, their unique minimum can be determined in $O(|L[s]|)$ time. Thus we can construct all best matches in $O(|L| + |L| \sum_s |L[s]|) = O(|L|^2)$ time.

THINNESS CLASSES. Recall that each connected component of a BMG $(\vec{G}, \sigma)$ has vertices with all $|S| \geq 2$ colors (we disregard the trivial case of the edge-less graph with $|S| = 1$) and thus, every $x \in V$ has a non-zero out-degree. Therefore $|\vec{E}| \geq |L|$, i.e., $O(|L| + |\vec{E}|) = O(|\vec{E}|) = O(|L|^2)$.

Consider a collection $\mathcal{F}$ of $n = |\mathcal{F}|$ subsets on L with a total size of $m = \sum_{A \in \mathcal{F}} |A|$. Then the set inclusion poset of $\mathcal{F}$ can be computed in $O(nm)$ time and $O(n^2)$ space as follows: For each $A \in \mathcal{F}$ run through all elements x of all other sets $B \in \mathcal{F}$ and mark $B \not\subseteq A$ if $x \notin A$, resulting in an $n \times n$ table $P_{\mathcal{F}}$ storing the set inclusion relation. More sophisticated algorithms that are slightly more efficient under particular circumstances are described in [185, 62].

In order to compute the thinness classes, we observe that the symmetric part of $P_{\mathcal{F}}$ corresponds to equal sets. The classes of equal sets can be obtained as connected components by Breadth-first search (BFS) [36] on the symmetric part of $P_{\mathcal{F}}$ with an effort of $O(n^2)$. This procedure is separately applied to the in- and out-neighborhoods of the BMG. Using an auxiliary graph in which $x, y \in L$ are connected if they are in the same component for both the in- and out- neighbors, the thinness classes can now be obtained by another BFS in $O(n^2)$. Since we have $n = |L|$ and $m = |\vec{E}|$, the sets of vertices with equal in- and out-neighborhoods can be identified in $O(|L|\,|\vec{E}|)$ total time.

RECOGNIZING 2-BMGS. Since Property (N0) holds for all 2-BMGs, it will be useful to construct the table X with entries $X_{\alpha,\beta} = 1$ if $\alpha \subseteq N^+(\beta)$ and $X_{\alpha,\beta} = 0$ otherwise. This table can be constructed in $O(|\vec{E}|)$ time by iterating over all edges and retrieving (in constant time) the R-classes to which its endpoints belong. The $N^+(N^+(\alpha))$ can now be obtained in $O(|\vec{E}|\,|L|)$ by iterating over all edges $\alpha\beta$ and adding the classes in $N^+(\beta)$ to $N^+(N^+(\alpha))$. We store this information in a table with entries $Q_{\alpha,\beta} = 1$ if $\alpha \in N^+(N^+(\beta))$ and $Q_{\alpha,\beta} = 0$ otherwise, in order to be able to decide membership in constant time later on.

A table $Y_{\alpha\beta}$ with $Y_{\alpha\beta} = 0$ if $N^+(\alpha) \cap N^+(N^+(\beta)) = \emptyset$ and $Y_{\alpha\beta} = 1$ if there is an overlap between $N^+(\alpha)$ and $N^+(N^+(\beta))$ can be computed in $O(|L|^3)$ time from the membership tables X and Q for neighborhoods $N^+(\,.\,)$ and next-

nearest neighborhoods $N^+(N^+(\,.\,))$, respectively. From the membership table for $N^+(N^+(\alpha))$ and $N^+(\gamma)$ we obtain $N^+(N^+(N^+(\alpha)))$ in $O(|\vec{E}|\,|L|)$ time, making use of the fact that $\sum_\alpha |N^+(\alpha)| = |\vec{E}|$. For fixed $\alpha, \beta \in \mathcal{N}$ it only takes constant time to check the conditions in (N1) and (N3) since all set inclusions and intersections can be tested in constant time using the auxiliary data derived above. The inclusion (N2) can be tested directly in $O(|L|)$ time for each α. We can summarize the considerations above as

Lemma 4.18. *A 2-BMG can be recognized in $O(|L|^2)$ space and $O(|L|^3)$ time with Algorithm 2.*

RECONSTRUCTION OF $T^*(\mathcal{H})$. For each $\alpha \in \mathcal{N}$, the reachable set $R(\alpha)$ can be found by a BFS in $O(|\vec{E}|)$ time and hence, with total complexity $O(|\vec{E}|\,|L|)$. For each α, we can find all $\beta \in \mathcal{N}$ with $N^-(\beta) = N^-(\alpha)$ and $N^+(\beta) \subseteq N^+(\alpha)$ in $O(|L|)$ time by simple look-ups in the set inclusion table for the in- and out-neighborhoods, respectively. Thus we can find all auxiliary leaf sets $Q(\alpha)$ in $O(|L^2|)$ time and the collection of the $R'(\alpha)$ can be constructed in $O(|\vec{E}|\,|L|)$.

The construction of the set inclusion poset is also useful to check whether the $\{R'(\alpha)\}$ form a hierarchy. In the worst case we have a tree of depth $|L|$ and thus, $m = O(|L|^2)$. Since the number of R-classes is bounded by $O(|L|)$, the inclusion poset of the reachable sets can be constructed in $O(|L|^3)$. The Hasse diagram of the partial order is the unique transitive reduction of the corresponding digraph. In our setting, this also takes $O(|L|^3)$ time [83, 2] since the inclusion poset of the $\{R'(\alpha)\}$ may have $O(|L|^2)$ edges. It is now easy to check whether the Hasse diagram is a tree or not. If the number of edges is at least the number of vertices, the answer is negative. Otherwise, the presence of a cycle can be verified e.g. using BFS in $O(|L|)$ time. It remains to check that the non-nested sets $R(\alpha)$ are indeed disjoint. It suffices to check this for the children of each vertex in the Hasse tree. Traversing the tree top-down this can be verified in $O(|L|^2)$ time since there are $O(|L|)$ vertices in the Hasse diagram and the total number of elements in the subtrees is $O(|L|)$.

Summarizing the discussion so far, and using the fact that the vertices $x \in \alpha$ can be attached to the corresponding vertices $R'(\alpha)$ in total time $O(|L|)$ we obtain

Lemma 4.19. *The unique least resolved tree $T^*(\mathcal{H}')$ of a connected 2-BMG $(\vec{G}, \sigma)$ can be constructed in $O(|L|^3)$ time and $O(|L|^2)$ space with Algorithm 2.*

INFORMATIVE TRIPLES. Since all informative triples $\mathcal{R}(\vec{G}, \sigma)$ come from an induced subgraph that contains at least one edge, it is possible to extract $\mathcal{R}(\vec{G}, \sigma)$ for a connected 2-BMG $(\vec{G} = (L, \vec{E}), \sigma)$ in $O(|\vec{E}|\,|L|)$ time. Furthermore, the total number of vertices and edges in $\mathcal{R}(\vec{G}, \sigma)$ is also bounded by $O(|\vec{E}|\,|L|)$, hence the algorithm of Deng and Fernández-Baca can be used to construct the tree $\mathrm{Aho}(\mathcal{R}(\vec{G}, \sigma))$ for a connected 2-BMG in $O(|\vec{E}|\,|L| \log^2(|\vec{E}|\,|L|))$ time [47]. The graph $(\vec{G}', \sigma)$ explained by this tree can be generated in $O(|L|^3)$ time, and checking whether $(\vec{G}, \sigma) = (\vec{G}', \sigma)$ requires $O(|L|^2)$ time. Asymptotically, the approach via informative triples (Algorithm 3) is therefore at best

as good as the direct construction of the least resolved tree $T^*(\mathcal{H}')$ with Algorithm 2.

EFFORT IN THE n-COLOR CASE. For n-BMGs it is first of all necessary to check all pairs of induced 2-BMGs. The total effort for processing all induced 2-BMGs is $O(\sum_{s<t}(|L[s]| + |L[t]|)^3) \leq O(|S|\,|L|\,\ell^2 + |L|^2\ell)$ with $\ell := \max_{s\in S} |L[s]|$, as shown by a short computation [1].

The 2-BMG for colors s and t is of size $O(L[s] + L[t])$ hence the total size of all $|S|(|S|-1)/2$ 2-BMGs is $O(|S|\,|L|)$. The total effort to construct a supertree from these 2-BMGs is therefore only $O(|L|\,|S|\log^2(|L|\,|S|))$ [47], and thus negligible compared to the effort of building the 2-BMGs.

Using Lemma 4.5 it is also possible to use the set of all informative triples directly. Its size is bounded by $O(|L|\,|\vec{E}|)$, hence the algorithm of Henzinger et al. [104] can be used to construct the supertree in $O(|L|\,|\vec{E}|\log^2(|L|\,|\vec{E}|))$. This bound is in fact worse than for the strategy of constructing all 2-BMGs first.

We note, finally, that for practical applications the number of genes between different species will be comparable, hence $O(\ell) = O(|L|/|S|)$. The total effort of recognizing an n-BMG in a biologically realistic application scenario amounts to $O(|L|^3/|S|)$. In the worst case scenario with $O(\ell) = O(|L|)$, the total effort is $O(|S|\,|L|^3)$.

4.6 SUMMARY

The main result of this chapter is a complete characterization of colored best match graphs (BMGs), a class of digraphs that arises naturally at the first stage of many of the widely used computational methods for orthology assignment. It has been shown that the problem of characterizing n-BMGs can be reduced to the less complex problem of characterizing 2-BMGs, and least resolved trees explaining a given n-BMG can be reconstructed by finding a supertree for the unique least resolved trees of all its induced 2-BMGs. In particular, any BMG $(\vec{G}, \sigma)$ is explained by a unique least resolved tree (T, σ), which is displayed by the true underlying tree. We have seen here that BMGs can be recognized in cubic time (in the number of genes) and with the same complexity it is possible to reconstruct the unique least resolved tree (T, σ). Related graph classes, for instance directed cographs [40], which appear in generalizations of orthology relations [100], or the Fitch graphs associated with horizontal gene transfer presented later in Chapter 8, have characterizations in terms of forbidden induced subgraphs. It seems quite likely that this not the case for best match graphs because they are not hereditary.

1 $O(\sum_{s<t}(|L[s]| + |L[t]|)^3) \leq O(\frac{1}{2}\sum_{s,t}(|L[s]| + |L[t]|)^3)$
$= O(\sum_{s,t}|L[s]^3| + 3\sum_{s,t}|L[s]|^2|L[t]|) \leq O(|S|\sum_s |L[s]|^3 + 3|L|\sum_s |L[s]|^2)$
$\leq O(|S|\,|L|\,\ell^2 + |L|^2\ell)$

5 RECIPROCAL BEST MATCH GRAPHS

Reciprocal best match graphs (RBMGs), i.e., the symmetric subgraphs of best match graphs, form the link between BMGs and orthology relations. While orthology is well-known to have a cograph structure [97, 96], this is in general not true for RBMGs (see e.g. Fig. 16). In fact, empirical observations (e.g. by Hellmuth et al. [98]) indicate that reciprocal best hit heuristics typically yield graphs with fairly large edit distances from cographs and thus from orthology relations. A complete characterization of RBMGs is therefore an indispensable prerequisite for the development of algorithms for the "RBMG-editing problem", i.e., the task to correct an empirically determined reciprocal best hit graph to a mathematically correct RBMG. Somewhat surprisingly, it turns out that this characterization is not a simple consequence of the results on BMGs.

This chapter is organized as follows: Section 5.1 formally introduces the reciprocal best match relation and gives a short motivation why the characterization of RBMGs cannot be derived in a straightforward manner from that of BMGs. In Section 5.2 the notion of least resolved trees is extended to RBMGs. However, as it turns out, such least resolved trees are not unique, in general. Complementary, Section 5.3 introduces a color-aware thinness relation S and shows that it suffices to characterize S-thin RBMGs. Combining these ideas, it is demonstrated in Section 5.4 that (G, σ) is an RBMG if and only if each of its connected components is an RBMG and at least one of them contains all colors, and give a simple construction for a tree explaining (G, σ) from trees for the connected components. In order to characterize connected, S-thin RBMGs, we first consider the case of three colors (Section 5.5). As we shall see, there are three distinct classes of 3-RBMGs that can be recognized in polynomial time. One of these classes does not contain induced paths on four vertices, while the other two classes do. In order to gain a better understanding of the two classes that contain induced P_4s, the influence of such P_4s is investigated in more detail in Section 5.6. This leads to three distinct types: the good, bad, and ugly P_4s. In Section 5.7 it will be proven that trees explaining an n-RBMG can be composed from tree-sets explaining the induced 3-RBMGs for all three-color subsets. However, the computational complexity for recognizing n-RBMGs is left as an open problem. Because of their practical relevance in orthology detection, this chapter is closed with a characterization of n-RBMGs that are cographs. As we shall see, the recognition of cograph n-RBMGs and the construction of trees that explain them can be done in polynomial time.
The results of this chapter have been published in Geiß et al. [75].

5.1 INTRODUCTION OF THE RECIPROCAL BEST MATCH RELATION

In this introductory section we formally define the reciprocal best match relation and its corresponding representation as an RBMG. Moreover, a short

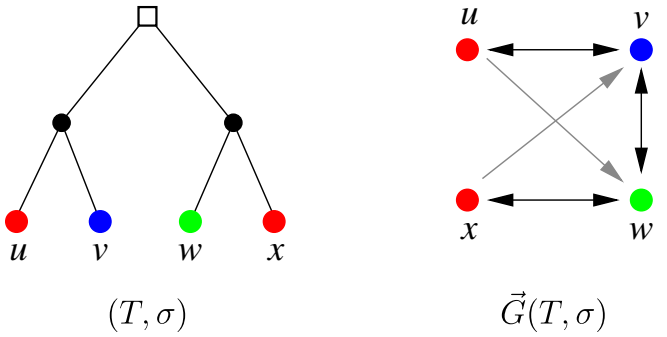


Fig. 16. Colored Reciprocal Best Match Graphs are not necessarily cographs. This simple counterexample shows a gene tree (T, σ), its corresponding BMG $\vec{G}(T, \sigma)$, and a species tree. The BMG contains the path (u, v, w, x) as symmetric part. The corresponding species tree (not shown here) is of the form (●(● ●)) with a duplication pre-dating the two speciations, with the speciation of ● and ● being followed by complementary loss of one of the t

motivation why the characterization of RBMGs is not a direct consequence of the properties of BMGs although both graphs are closely related, is given at the end of the section.
Reciprocal best matches naturally arise from the definition of best matches:

Definition 5.1. *If the leaf y is a* best match *of the leaf x in the gene tree T and x is also a best match of y, we call x and y* reciprocal best matches.

The reciprocal best match relation is symmetric by definition. As the best match relation, it is reflexive because every gene x in species s is its own (unique) best match within s.
The reciprocal best match relation is conveniently represented as a vertex-colored undirected graph (G, σ) with vertex set L whose edges represent reciprocal best matches. Similarly to best match graphs, we can again consider (G, σ) as loop-less. The relationship between RBMGs and the trees from which they are derived is captured by

Definition 5.2. *Given a tree T and a map $\sigma : L \to S$, the* colored reciprocal best match graph *(RBMG) $G(T, \sigma)$ has vertex set L and edges $xy \in E(G)$ if x and y are reciprocal best matches and $x \neq y$. Each vertex $x \in L$ obtains the color $\sigma(x)$.*
The rooted tree T explains *the vertex-colored graph (G, σ) if (G, σ) is isomorphic to the RBMG $G(T, \sigma)$.*

In analogy to BMGs, we will often speak of $|S|$-RBMGs to refer to the number if $|S|$ colors. Moreover, Def. 5.2 immediately implies

Observation 5.1. *If (G, σ) is an RBMG, then σ is a proper vertex coloring.*

As a consequence, (G, σ) cannot be explained by a leaf-colored tree unless σ is a proper vertex coloring. As in the context of BMGs, we therefore assume throughout this chapter that (G, σ) is properly colored.
By definition $(G = (V, E), \sigma)$ is an RBMG if and only if there is a BMG $(\vec{G}', \sigma)$ with vertex set V such that $xy \in E(G)$ if and only if both (x, y) and

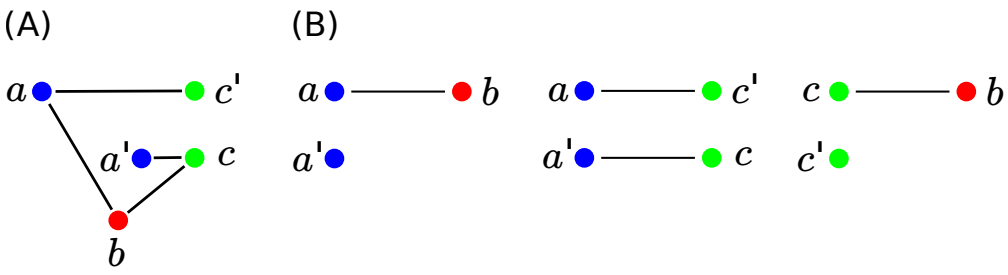


Fig. 17. (A) A symmetric graph on three colors. (B) Each induced subgraph on two colors is a reciprocal best match graph and a disjoint union of complete bipartite graphs. However, the corresponding symmetric graph on three colors shown in (A) does not have a tree representation.

(y, x) are arcs in $(\vec{G}', \sigma)$. In particular, therefore, an RBMG is the edge-disjoint union of the edge sets of the induced RBMGs by pairs of distinct colors $s, t \in S$.

Corollary 5.1. *Every 2-RBMG is the disjoint union of complete bipartite graphs.*

Proof. Let (G, σ) be a 2-RBMG that is contained as symmetric part in some 2-BMG $(\vec{G}, \sigma)$. By Lemma 4.4, there are arcs (x, y) and (y, x) in $\vec{G}$ if and only if $x \in \alpha \subseteq N^+(\beta)$ and $y \in \beta \subseteq N^+(\alpha)$ for distinct R-classes α, β. In this case $\rho_\alpha = \rho_\beta$. By Lemma 4.3(v), it then holds $\sigma(\alpha) \neq \sigma(\beta)$. The same results also implies that in a 2-RBMG there are at most two R-classes with the same root. Thus the connected components of a 2-RBMG are the complete bipartite graphs formed by pairs of R-classes with a common root, as well as isolated vertices corresponding to all other leaves of T.

□

The converse, however, is not true, as shown by the counterexample in Fig. 17. This example in particular suggests that, in contrast to BMGs, the characterization of an RBMG cannot be broken down to its 2-colored induced subgraphs. In fact, the complete characterization of RBMGs does not seem to follow in a straightforward manner from the properties of the underlying BMGs.

5.2 LEAST RESOLVED TREES

Before deriving a complete characterization of RBMGs, we derive in this section the notion of least resolved trees in the context of RBMGs. As we shall see below, the characterization of these trees is closely related to the one of best matches but cannot be expressed in terms of *reciprocal* best matches alone.

Given a leaf-colored tree (T, σ), one can easily derive the respective BMG $\vec{G}(T, \sigma)$ and RBMG $G(T, \sigma)$ that are explained by (T, σ). Conversely, if (G, σ) is an RBMG, then there is a tree (T, σ) that explains (G, σ). This tree also explains the digraph $\vec{G}(T, \sigma)$ with the property that $xy \in E(G)$ if and only if both (x, y) and (y, x) are arcs in $\vec{G}(T, \sigma)$. A colored graph (G, σ) therefore is an RBMG if and only if it is the symmetric part of some BMG.

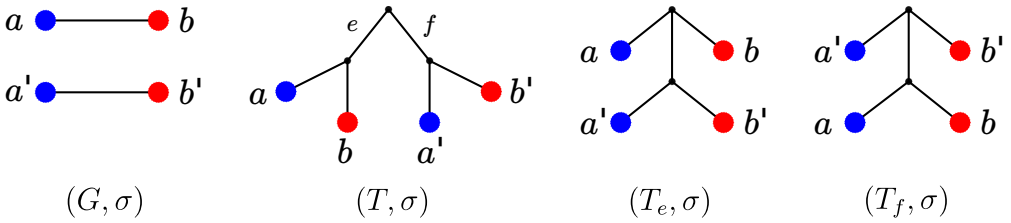


Fig. 18. The reciprocal best match graph (G,σ) on two colors (red and blue) is explained by (T,σ) which contains the redundant edges e and f. Contraction of one of these edges gives (T_e,σ) and (T_f,σ), respectively, which are both least resolved but distinct from each other, i.e., there exists no unique least resolved tree w.r.t. (G,σ). In particular, none of the trees (T_{ef},σ) and (T_{fe},σ) explains (G,σ).

It is important to note that there can be distinct trees (T',σ) and (T'',σ) that explain the same RBMG, i.e., $G(T',\sigma) = G(T'',\sigma)$, albeit the leaf set L and the leaf coloring σ of course must be the same. In general the BMGs $\vec{G}(T',\sigma)$ and $\vec{G}(T'',\sigma)$ can also be different, even if $G(T',\sigma) = G(T'',\sigma)$. For an example, consider the RBMG (G,σ) and the two distinct trees (T_e,σ) and (T_f,σ) in Fig. 18. We have $(G,\sigma) = G(T_e,\sigma) = G(T_f,\sigma)$. However, $\vec{G}(T_e,\sigma)$ contains the arc (a,b') which is not contained in $\vec{G}(T_f,\sigma)$. Hence, $\vec{G}(T_e,\sigma) \neq \vec{G}(T_f,\sigma)$.

Although there are in general many different trees that explain the same BMG or RBMG, we have already seen in Chapter 4 that every best match graph $(\vec{G},\sigma)$ is explained by a uniquely defined "smallest" tree, its so-called *least resolved* tree. Recall that least resolved trees in the BMG setting are intimately related to roots of R-classes. The notion of least resolved trees is also of interest for RBMGs even though we shall see below that they are not unique in the reciprocal setting.

Definition 5.3. *Let (G,σ) be an RBMG that is explained by a tree (T,σ). An inner edge e is called* redundant *if (T_e,σ) also explains (G,σ), otherwise e is called* relevant.

The next result gives a characterization of redundant edges:

Lemma 5.1. *Let (G,σ) be an RBMG explained by (T,σ). An inner edge $e = uv$ in T is redundant if and only if it satisfies*

(LR) *For all colors $s \in \sigma(L(T(v))) \cap \sigma(L(T(u)) \setminus L(T(v)))$, it holds that if $v = \rho_{\alpha,s}$ for some R-class $\alpha \in \mathcal{N}(\vec{G}(T,\sigma))$, then $\rho_{\beta,\sigma(\alpha)} \prec u$ for every R-class $\beta \subseteq L(T(u)) \setminus L(T(v))$ of $\vec{G}(T,\sigma)$ with $\sigma(\beta) = s$.*

Proof. The R-classes appearing throughout this proof refer to the directed graph $(\vec{G},\sigma) = \vec{G}(T,\sigma)$ and hence, are completely determined by (T,σ). By definition, any redundant edge of (T,σ) is an inner edge, thus we can assume that $e = uv$ is an inner edge of (T,σ) throughout the whole proof.

Suppose that Property (LR) is satisfied. We show (with the help of Lemma 4.14) that most neighborhoods in the BMG $(\vec{G},\sigma) := \vec{G}(T,\sigma)$ remain unchanged by the contraction of e, while those neighborhoods that change do so in such a way that (T_e,σ) still explains the RBMG (G,σ).

We denote the inner vertex in T_e obtained by contracting $e = uv$ again by u. Recall that by convention $u \succ_T v$ in T. By construction, we have $L(T(w)) = L(T_e(w))$ for all $w \neq v$ and $\mathrm{lca}_T(x, y) = \mathrm{lca}_{T_e}(x, y)$ unless $\mathrm{lca}_T(x, y) = v$. Hence, a root $\rho_{\alpha,s} \neq v$ of (T, σ) is also a root in (T_e, σ). Lemma 4.14 thus implies that $N_s^+(\alpha)$ remains unchanged upon contraction of e whenever $\rho_{\alpha,s} \neq v$.

Now let α and s be such that $v = \rho_{\alpha,s}$, thus $N_s^+(\alpha) = L(T(v)) \cap L[s]$ by Lemma 4.14 and in particular $s \in \sigma(L(T(v)))$. We distinguish two cases:
(1) If $s \notin \sigma(L(T(u)) \setminus L(T(v)))$, then there is no R-class $\beta \subseteq L(T(u)) \setminus L(T(v))$ of color s, which implies $L(T(u)) \cap L[s] = L(T(v)) \cap L[s]$. Hence, the set $N_s^+(\alpha)$ remains unaffected by contraction of e.
(2) Assume $s \in \sigma(L(T(u)) \setminus L(T(v)))$ and let $\beta \subseteq L(T(u)) \setminus L(T(v))$ be an R-class of color $\sigma(\beta) = s$. Moreover, let $\sigma(\alpha) = r \neq s$. We thus have $\rho_{\beta,r} \prec_T u$ by Property (LR). Now, $N_s^+(\alpha) = L(T(v)) \cap L[s]$ and $\beta \subseteq L(T(u)) \setminus L(T(v))$ imply $\beta \cap N_s^+(\alpha) = \emptyset$. Moreover, Lemma 4.14 and $\rho_{\beta,r} \prec_T u$ imply $\alpha \cap N_r^+(\beta) = \emptyset$ in (T, σ), i.e., $xy \notin E(G)$ for any $x \in \alpha$ and $y \in \beta$ since neither (x, y) nor (y, x) is an arc in $\vec{G}$. After contraction of e, we have $\rho_{\beta,r} \prec \rho_{\alpha,s}$, i.e., $\beta \subseteq N_s^+(\alpha)$, but $\alpha \cap N_r^+(\beta) = \emptyset$ in (T_e, σ) by Lemma 4.14. Thus we have $(x, y) \in E(\vec{G})$ and $(y, x) \notin E(\vec{G})$, which implies $xy \notin E(G(T_e, \sigma))$. In summary, we can therefore conclude that (T_e, σ) still explains (G, σ).

Conversely, suppose that e is a redundant edge. If there is no R-class α with $v = \rho_{\alpha,s}$, then Lemma 4.14 again implies that contraction of e does not affect the out-neighborhoods of any R-classes, thus (T_e, σ) explains (G, σ). Hence, assume for contradiction that there is a color $s \in \sigma(L(T(v))) \cap \sigma(L(T(u)) \setminus L(T(v)))$ and an R-class $\beta \subseteq L(T(u)) \setminus L(T(v))$ of color s with $\rho_{\beta,r} \succeq u$, where $r \in S \setminus \{s\}$ such that there exists an R-class α of color $\sigma(\alpha) = r$ with $v = \rho_{\alpha,s}$. Note that this in particular means that there is no leaf z of color r in $L(T(u)) \setminus L(T(v))$ as otherwise $\mathrm{lca}(\beta, z) \prec_T u = \rho_{\beta,r} = \mathrm{lca}(\beta, \alpha)$; a contradiction since $\alpha \in N_r^+(\beta)$ by Lemma 4.14. As $\alpha \prec v$ by construction, we have $u = \mathrm{lca}(\alpha, \beta)$ and therefore $\rho_{\beta,r} = u$. In particular, it holds $\rho_{\beta,r} \succ \rho_{\alpha,s}$. As a consequence, we have $\beta \cap N_s^+(\alpha) = \emptyset$ and $\alpha \subseteq N_r^+(\beta)$ in (T, σ), again by Lemma 4.14. Thus, for any $x \in \alpha$ and $y \in \beta$, we have $(x, y) \notin E(\vec{G})$ and $(y, x) \in E(\vec{G})$ and therefore, $xy \notin E(G)$. Since $\rho_{\beta,r} = u$, contraction of e implies $\rho_{\beta,r} = \rho_{\alpha,s}$ in (T_e, σ). Therefore $(x, y) \in E(\vec{G})$ and $(y, x) \in E(\vec{G})$, which implies $xy \in E(G(T_e, \sigma))$. Thus (T_e, σ) does not explain (G, σ); a contradiction. □

It is interesting to note that the characterization of redundancy (w.r.t. an RBMG) of edges in (T, σ) requires information on (directed) best matches and apparently cannot be expressed entirely in terms of the reciprocal best match relation. In particular, Property (LR) requires R-classes.

The next result provides alternative sufficient conditions for least resolved trees. In particular, it shows whether inner edges uv can be contracted based on the particular colors of leaves below the children of u. We will show in the last section that the conditions in Lemma 5.2 are also necessary for RBMGs that are cographs (cf. Lemma 5.45). These conditions are thus designed to fit in well within the framework of RBMGs that are cographs, which will be introduced in more detail later, although these conditions may be relaxed for the general case.

Lemma 5.2. *Let (G,σ) be an RBMG explained by (T,σ) and let $e = uv$ be an inner edge of T. Moreover, for two vertices x, y in T, we define $S_{x,\neg y} := \sigma(L(T(x))) \setminus \sigma(L(T(y)))$. Then (T_e,σ) explains (G,σ) if one of the following conditions is satisfied:*

(1) $\sigma(L(T(v'))) \cap \sigma(L(T(v))) = \emptyset$ *for all* $v' \in \mathsf{child}_T(u)$*, or*

(2) $\sigma(L(T(v'))) \cap \sigma(L(T(v))) \in \{\sigma(L(T(v))), \sigma(L(T(v')))\}$ *for all* $v' \in \mathsf{child}_T(u)$*, and either*

  (i) $\sigma(L(T(v))) \subseteq \sigma(L(T(v')))$ *for all* $v' \in \mathsf{child}_T(u)$*, or*

  (ii) if $\sigma(L(T(v'))) \subsetneq \sigma(L(T(v)))$ *for some* $v' \in \mathsf{child}_T(u)$*, then, for every* $w \in \mathsf{child}_T(v)$ *that satisfies* $S_{w,\neg v'} \neq \emptyset$*, it holds that* $\sigma(L(T(v')))$ *and* $\sigma(L(T(w)))$ *do not overlap and thus,* $\sigma(L(T(v'))) \subseteq \sigma(L(T(w)))$*.*

Proof. Suppose that $e = uv$ satisfies one of the Properties (1) or (2). If Property (1) is satisfied, we clearly have $\sigma(L(T(v))) \cap \sigma(L(T(u)) \setminus L(T(v))) = \emptyset$, which implies that Condition (LR) of Lemma 5.1 is trivially satisfied. Therefore e is redundant in (T,σ) and, by Def. 5.3, (T_e,σ) explains (G,σ).

Now let $\sigma(L(T(v'))) \cap \sigma(L(T(v))) \in \{\sigma(L(T(v))), \sigma(L(T(v')))\}$ for all $v' \in \mathsf{child}_T(u)$ and assume that either Property (2.i) or (2.ii) is satisfied. In order to see that (T_e,σ) explains (G,σ), we show that e is redundant in (T,σ) by application of Lemma 5.1. Thus suppose $v = \rho_{\alpha,s}$ for some R-class $\alpha \in \mathcal{N}(\vec{G}(T,\sigma))$. If there exists no R-class $\beta \subseteq L(T(u)) \setminus L(T(v))$ of $\vec{G}(T,\sigma)$ with $\sigma(\beta) = s$, then Lemma 5.1 is again trivially satisfied and (T_e,σ) explains (G,σ). Hence, suppose that there is an R-class $\beta \subseteq L(T(u)) \setminus L(T(v))$ of $\vec{G}(T,\sigma)$ with $\sigma(\beta) = s$. Clearly, if $\beta \preceq_T x \prec_T u$ for some $x \in \mathsf{child}_T(u) \setminus \{v\}$ with $\sigma(L(T(v))) \subseteq \sigma(L(T(x)))$, then $\rho_{\beta,\sigma(\alpha)} \preceq_T x \prec_T u$.

Hence, if Property (2.i) holds, i.e., $\sigma(L(T(v))) \subseteq \sigma(L(T(v')))$ for all $v' \in \mathsf{child}_T(u)$, we easily see that for all R-classes $\beta \subseteq L(T(u)) \setminus L(T(v))$ with $\sigma(\beta) = s$ we have $\rho_{\beta,\sigma(\alpha)} \preceq_T x \prec_T u$ for some $x \in \mathsf{child}_T(u) \setminus \{v\}$. Therefore e is redundant in (T,σ) and (T_e,σ) explains (G,σ).

Now suppose that Property (2.ii) holds. If $\sigma(\alpha) \in \sigma(L(T(v')))$ for each $v' \in \mathsf{child}_T(u)$, we easily see that $\rho_{\beta,\sigma(\alpha)} \preceq_T x \prec_T u$ for some $x \in \mathsf{child}_T(u) \setminus \{x\}$. Otherwise, there exists some $\tilde{v} \in \mathsf{child}_T(u) \setminus \{v\}$ such that $\sigma(\alpha) \notin \sigma(L(T(\tilde{v})))$. By Property (2), $\sigma(L(T(\tilde{v})))$ and $\sigma(L(T(v)))$ do not overlap. Therefore $\sigma(L(T(\tilde{v}))) \subsetneq \sigma(L(T(v)))$. In order to show that (LR) is satisfied, we thus need to show that $s \notin \sigma(L(T(\tilde{v})))$, otherwise $\rho_{\beta',s} = u$ for some R-class $\beta' \subseteq L(T(u)) \setminus L(T(v))$ of $\vec{G}(T,\sigma)$. Let $w \in \mathsf{child}_T(v)$ such that $a \preceq_T w$ for some $a \in \alpha$. Since $\sigma(\alpha) \notin \sigma(L(T(\tilde{v})))$, it follows $S_{w,\neg\tilde{v}} \neq \emptyset$. Hence, by Property (2.ii), it must hold $\sigma(L(T(\tilde{v}))) \subseteq \sigma(L(T(w)))$. Since $\rho_{\alpha,s} = v$ by assumption, we necessarily have $s \notin \sigma(L(T(w)))$ and thus, as $\sigma(L(T(\tilde{v}))) \subseteq \sigma(L(T(w)))$, we can conclude $s \notin \sigma(L(T(\tilde{v})))$. Thus, for all children $v' \in \mathsf{child}_T(u)$, we either have $\sigma(\alpha) \in \sigma(L(T(v')))$ or $\sigma(\alpha), \sigma(\beta) \notin \sigma(L(T(v')))$. Now, one can easily see that $\rho_{\beta,\sigma(\alpha)} \preceq_T x \prec_T u$ for some $x \in \mathsf{child}_T(u) \setminus \{x\}$. Hence, Condition (LR) from Lemma 5.1 is always satisfied. Therefore the edge e is redundant in (T,σ), i.e., (T_e,σ) explains (G,σ).

□

Definition 5.4. *Let (G,σ) be an RBMG explained by (T,σ). Then (T,σ) is* least resolved w.r.t. *(G,σ) if (T_A,σ) does not explain (G,σ) for any non-empty series of edges A of (T,σ).*

Again, the reference to the explicit reference graph (G,σ) will be dropped whenever the context is clear. In particular, least resolved trees in this chapter are always considered w.r.t. an RBMG unless explicitly stated otherwise.

Given two distinct redundant edges $e \neq f$ of (T,σ), the edge f is not necessarily redundant in (T_e,σ), i.e., the tree (T_{ef},σ) obtained by sequential contraction of e and f does not necessarily explain (G,σ). This in particular implies that the contraction of all redundant edges of (T,σ) does not necessarily result in a least resolved tree for the same RBMG. Moreover, there may be more than one least resolved tree that explains a given n-RBMG (G,σ). Fig. 18 gives an example of least resolved trees that are not unique.
The results about least resolved trees are summarized in the following

Theorem 5.1. *Let (G,σ) be an RBMG explained by (T,σ). Then there exists a (not necessarily unique) least resolved tree $(T_{e_1\dots e_k},\sigma)$ explaining (G,σ) obtained from (T,σ) by a series of edge contractions $e_1e_2\dots e_k$ such that the edge e_1 is redundant in (T,σ) and e_{i+1} is redundant in $(T_{e_1\dots e_i},\sigma)$ for $i \in \{1,\dots,k-1\}$. In particular, (T,σ) displays $(T_{e_1\dots e_k},\sigma)$.*

Proof. The Theorem follows directly from the definition of least resolved trees and the observation that for any two redundant edges $e \neq f$ of (T,σ), the tree (T_{ef},σ) does not necessarily explain (G,σ). Clearly, by definition, $(T_{e_1\dots e_k},\sigma)$ is displayed by (T,σ).

□

5.3 S-THINNESS

The R relation introduced in the previous chapter is the natural generalization of thinness in undirected graphs [159]. As already argued in Chapter 4, all vertices within an R-class of a BMG have the same color. However, a corresponding result does not hold for RBMGs. Fig. 19 shows a counterexample, where $N(a) = N(b)$ holds for vertices with different colors $\sigma(a) \neq \sigma(b)$. Since color plays a key role in our context, we introduce a color-preserving thinness relation:

Definition 5.5. *Let (G,σ) be an undirected colored graph. Then two vertices a and b are in relation* S*, in symbols $a\mathsf{S}b$, if $N(a) = N(b)$ and $\sigma(a) = \sigma(b)$. An undirected colored graph (G,σ) is* S*-thin if no two distinct vertices are in relation* S*. We denote the* S*-class that contains the vertex x by $[x]$.*

As all elements within an R-class have the same color in a BMG and every RBMG (G,σ) is the symmetric part of some BMG $\vec{G}(T,\sigma)$, we obtain

Lemma 5.3. *Let (G,σ) be an RBMG, (T,σ) a tree explaining (G,σ), and $\vec{G}(T,\sigma)$ the corresponding BMG. Then $x\mathsf{R}y$ in $\vec{G}(T,\sigma)$ implies that $x\mathsf{S}y$ in (G,σ).*

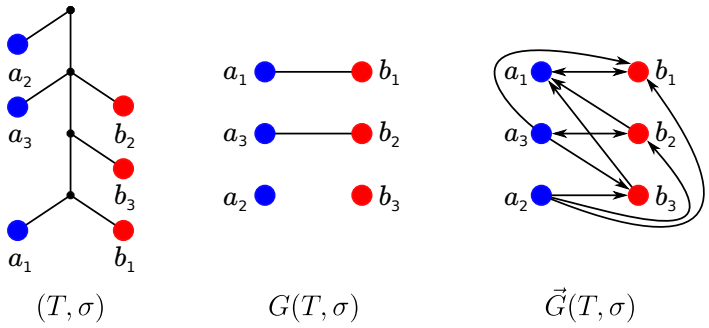


Fig. 19. The leaf-colored tree (T, σ) on the left explains the RBMG $G(T, \sigma)$ (middle) and the BMG $\vec{G}(T, \sigma)$ (right). The colored graph $\vec{G}(T, \sigma)$ is R-thin. Thus all leaves within an R-class are trivially of the same color. However, in the RBMG we have $N(a_2) = N(b_3) = \emptyset$ but a_2 and b_3 are of different color. Note, by definition, a_2 and b_3 are not within the same S-class.

The converse of Lemma 5.3 is not true, however. In Fig. 19, for instance, changing the color of the leaf b_3 from blue to red in the tree (T, σ) implies $N(a_2) = N(b_3)$ in the RBMG (G, σ) and the set $\{a_2, b_3\}$ forms an S-class. On the other hand, we have $N^+(a_2) \neq N^+(b_3)$ in the corresponding BMG $\vec{G}(T, \sigma)$, thus a_2 and b_3 do not belong to the same R-class of $\vec{G}(T, \sigma)$.

For an undirected colored graph (G, σ), we denote by G/S the graph whose vertex set are exactly the S-classes of (G, σ), and two distinct classes $[x]$ and $[y]$ are connected by an edge in G/S if there is an $x' \in [x]$ and $y' \in [y]$ with $x'y' \in E(G)$. Moreover, since the vertices within each S-class have the same color, the map $\sigma_{/\mathsf{S}} \colon V(G/\mathsf{S}) \to S$ with $\sigma_{/\mathsf{S}}([x]) = \sigma(x)$ is well-defined.

Lemma 5.4. *$(G/\mathsf{S}, \sigma_{/\mathsf{S}})$ is S-thin for every undirected colored graph (G, σ). Moreover, $xy \in E(G)$ if and only if $[x][y] \in E(G/\mathsf{S})$. Thus G is connected if and only if G/S is connected.*

Proof. First, we show that $xy \in E(G)$ if and only if $[x][y] \in E(G/\mathsf{S})$. Assume $xy \in E(G)$. Since G does not contain loops, we have $x \notin N(x)$. However, $x \in N(y)$. Therefore $N(x) \neq N(y)$ and thus, $[x] \neq [y]$. By definition, thus, $[x][y] \in E(G/\mathsf{S})$.

Now assume $[x][y] \in E(G/\mathsf{S})$. By construction, there exists $x' \in [x]$ and $y' \in [y]$ such that $x'y' \in E(G)$ and thus $x' \in N(y') = N(y)$ and $y' \in N(x') = N(x)$. In particular, $x'y' \in E(G)$ implies $\sigma(x') \neq \sigma(y')$ and thus $\sigma(x) \neq \sigma(y)$ since by definition all vertices within an S-class are of the same color. Therefore $xy \in E(G)$ by definition of S-thinness.

Now suppose, for contradiction, that $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$ is not S-thin. Then there are two distinct vertices $[x], [y]$ in G/S that have the same neighbors $[v_1], \ldots, [v_k]$ in G/S and $\sigma_{/\mathsf{S}}([x]) = \sigma_{/\mathsf{S}}([y])$ and, in particular, $\sigma(x) = \sigma(y)$. From "$xy \in E(G)$ if and only if $[x][y] \in E(G/\mathsf{S})$" we infer $N_G(x) = \bigcup_{i=1}^{k} \bigcup_{v \in [v_i]} \{v\} = N_G(y)$ and thus $[x] = [y]$; a contradiction. Thus $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$ must be S-thin. □

The map $\gamma_{\mathsf{S}} \colon V(G) \to V(G/\mathsf{S}) : x \mapsto [x]$ collapses all elements of an S-class in (G, σ) to a single node in $(G_{/\mathsf{S}}, \sigma_{/\mathsf{S}})$. Hence, the γ_{S}-image of a connected component of (G, σ) is a connected component in $(G_{/\mathsf{S}}, \sigma_{/\mathsf{S}})$. Conversely, the

pre-image of a connected component of $(G_{/\mathsf{S}}, \sigma_{/\mathsf{S}})$ that contains an edge is a single connected component of (G, σ). Furthermore, $(G_{/\mathsf{S}}, \sigma_{/\mathsf{S}})$ contains at most one isolated vertex of each color $r \in S$. If it exists, then its pre-image is the set of all isolated vertices of color r in (G, σ); otherwise (G, σ) has no isolated vertex of color r. The next lemma shows how a tree (T, σ) that explains an RBMG (G, σ) can be modified to a tree that still explains (G, σ) by replacing edges that are connected to vertices within the same S-class. Although this lemma is quite intuitive, one needs to be careful in the proof since changing edges in (T, σ) may also change the neighborhoods $N_G(x)$ of vertices $x \in V(G)$ and may result in a tree that does not explain (G, σ) anymore.

Lemma 5.5. *Let (G, σ) be an RBMG that is explained by (T, σ) on L. Let $x, x' \in [x]$ be two distinct vertices in an S-class $[x]$ of (G, σ). Suppose that x and x' have distinct parents v_x and $v_{x'}$ in T, respectively. Denote by T_{x',v_x} the tree on L obtained from T by (i) removing the edge $(v_{x'}, x')$, (ii) suppressing the vertex $v_{x'}$ if it now has degree 2, and (iii) inserting the edge (v_x, x'). Then, (T_{x',v_x}, σ) explains (G, σ).*

Proof. Let $[x]$ be an S-class with vertices $x, x' \in [x]$ that have distinct parents v_x and $v_{x'}$ in T, respectively. Put $T' = T_{x',v_x}$ and let (G', σ) be the RBMG explained by (T', σ). We proceed with showing that $(G', \sigma) = (G, \sigma)$. To see this, we observe that $x, x' \in [x]$ implies that $N_G(x) = N_G(x')$ and $\sigma(x) = \sigma(x')$. By construction, we also have $N_{G'}(x) = N_{G'}(x')$ and $x' \notin N_{G'}(x)$. Moving x' in T does not affect the last common ancestors of x and any $y \neq x'$, hence $N_{G'}(x) = N_G(x)$ and thus, also $N_{G'}(x') = N_G(x)$. Now consider $N_{G'}(y)$ and $N_G(y)$ for some $y \neq x, x'$ and assume, for contradiction, that $N_{G'}(y) \neq N_G(y)$. Then there exists a vertex $z \in N_G(y) \setminus N_{G'}(y)$ or $z \in N_{G'}(y) \setminus N_G(y)$, which in particular implies $N_G(z) \neq N_{G'}(z)$. As shown above, $N_{G'}(x) = N_G(x) = N_G(x') = N_{G'}(x')$. Hence, $N_G(z) \neq N_{G'}(z)$ implies $z \neq x, x'$. Moreover, since z is adjacent to y in either G or G', we have $\sigma(z) \neq \sigma(y)$. However, replacing x' in T cannot influence the adjacencies between vertices u and v with $\sigma(u) \neq \sigma(x')$ and $\sigma(v) \neq \sigma(x')$. Taken the latter arguments together, we can conclude that $\sigma(z) = \sigma(x) \neq \sigma(y)$.
First assume $z \in N_G(y) \setminus N_{G'}(y)$. Then

$$\mathrm{lca}_T(z, y) \preceq_T \mathrm{lca}_T(z', y) \text{ for all } z' \text{ with } \sigma(z') = \sigma(z) \text{ and} \tag{9}$$
$$\mathrm{lca}_T(z, y) \preceq_T \mathrm{lca}_T(z, y') \text{ for all } y' \text{ with } \sigma(y') = \sigma(y). \tag{10}$$

Since $z \notin N_{G'}(y)$, we additionally have

$$\mathrm{lca}_{T'}(z, y) \succ_{T'} \mathrm{lca}_{T'}(z', y) \text{ for some } z' \text{ with } \sigma(z') = \sigma(z) \text{ or} \tag{11}$$
$$\mathrm{lca}_{T'}(z, y) \succ_{T'} \mathrm{lca}_{T'}(z, y') \text{ for some } y' \text{ with } \sigma(y') = \sigma(y). \tag{12}$$

The fact that T and T' are identical up to the location of x' together with $\sigma(z') = \sigma(x') \neq \sigma(y)$ and $x' \neq z$ implies that in T' we still have $\mathrm{lca}_{T'}(z, y) \preceq_{T'} \mathrm{lca}_{T'}(z, y')$ for all y' with $\sigma(y') = \sigma(y)$. Hence, Equ. (11) must be satisfied. Equ. (9) and (11) together imply that $x' = z'$ and that x' is the only vertex that satisfies Equ. (11). In T' the vertices x and x' have the same parent. Together with $x' = z'$ and Equ. (11) this implies $\mathrm{lca}_{T'}(x, y) = \mathrm{lca}_{T'}(x', y) \prec_{T'} \mathrm{lca}_{T'}(z, y)$.

Since T and T' are identical up to the location of x', we also have $\mathrm{lca}_{T'}(x,y) = \mathrm{lca}_T(x,y)$ and $\mathrm{lca}_{T'}(y,z) = \mathrm{lca}_T(y,z)$. Combining these arguments, we obtain $\mathrm{lca}_T(x,y) \prec_T \mathrm{lca}_T(y,z)$, which contradicts Equ. (9) because $\sigma(z) = \sigma(x)$.
Assuming $z \in N_{G'}(y) \setminus N_G(y)$ and interchanging the role of T and T' in the argument above, we obtain

$$\mathrm{lca}_T(z,y) \succ_T \mathrm{lca}_T(x',y) \text{ and} \tag{13}$$
$$\mathrm{lca}_T(z,y) \preceq_T \mathrm{lca}_T(z,y') \text{ for all } y' \text{ with } \sigma(y') = \sigma(y) \tag{14}$$

and that there is no other vertex $z^* \neq x'$ with $\sigma(z^*) = \sigma(x')$ and $\mathrm{lca}_T(z,y) \succ_T \mathrm{lca}_T(z^*,y)$. Since x and x' have the same parent in T', we have $\mathrm{lca}_T(x,y) = \mathrm{lca}_{T'}(x,y) = \mathrm{lca}_{T'}(x',y) \succeq_{T'} \mathrm{lca}_{T'}(z,y) = \mathrm{lca}_T(z,y) \succ_T \mathrm{lca}_T(x',y)$. The fact that T and T' are identical up to the location of x' now implies that for all inner vertices v, w of T' we have $v \prec_{T'} w$ if and only if $v \prec_T w$. Hence, we have

$$\mathrm{lca}_T(x,y) \succeq_T \mathrm{lca}_T(z,y) \succ_T \mathrm{lca}_T(x',y)$$

implying that T displays the triple $x'y|x$. Therefore xy is not an edge in (G,σ), whence $y \notin N_G(x) = N_G(x')$.
Since there is no other vertex $z^* \neq x'$ with $\sigma(z^*) = \sigma(x')$ and $\mathrm{lca}_T(z,y) \succ_T \mathrm{lca}_T(z^*,y)$, we have $\mathrm{lca}_T(z^*,y) \succ_T \mathrm{lca}_T(x',y)$ for all $z^* \neq x'$ with $\sigma(z^*) = \sigma(z) = \sigma(x')$. Since $y \notin N_G(x')$, there must be a vertex y' with $\sigma(y') = \sigma(y)$ such that $\mathrm{lca}_T(x',y) \succ_T \mathrm{lca}_T(x',y')$. We can choose y' such that there is no other vertex $y^* \neq y'$ satisfying $\sigma(y^*) = \sigma(y')$ and $\mathrm{lca}_T(x',y') \succ_T \mathrm{lca}_T(x',y^*)$. Thus we have

$$\mathrm{lca}_T(x',y') \prec_T \mathrm{lca}_T(x',y) \prec_T \mathrm{lca}_T(x,y),$$

which implies $y' \notin N_G(x)$. However, since x' is unique w.r.t. Equ. (13), we must have $y' \in N_G(x')$; a contradiction to $N_G(x) = N_G(x')$.
Therefore we have $N_G(v) = N_{G'}(v)$ for all $v \in V(G)$ and thus, $(G,\sigma) = (G',\sigma)$ as claimed. □

The following result sheds some more light on the relationship of an RBMG (G,σ) and its corresponding S-thin version $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$. It in particular shows how a tree explaining (G,σ) can be obtained from a tree explaining $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$.

Lemma 5.6. *(G,σ) is an RBMG if and only if $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$ is an RBMG. Moreover, every RBMG (G,σ) is explained by a tree $(\widehat{T},\sigma)$ in which any two vertices $x, x' \in [x]$ of each S-class $[x]$ of (G,σ) have the same parent.*

Proof. Consider an RBMG (G,σ) explained by the tree (T,σ), and let $[x]$ be an S-class of (G,σ). If all the vertices within $[x]$ have the same parent v in T, then we can identify the edges vx' for all $x' \in [x]$ to obtain the edge $v[x]$. If all children of v are leaves of the same color, we additionally suppress v in order to obtain a phylogenetic tree $T/[x]$. Note that in this case, $\mathsf{par}(v)$ cannot be incident to any leaf y of color $\sigma(x)$ in (T,σ) as this would imply $N(x) = N(y)$ and therefore $x\mathsf{S}y$. Hence, suppression of v has no effect on any of the neighborhoods and thus, does not affect any of the reciprocal best matches in T. If all S-classes are of this form, then the tree $(T/\mathsf{S}, \sigma_{/\mathsf{S}})$ obtained

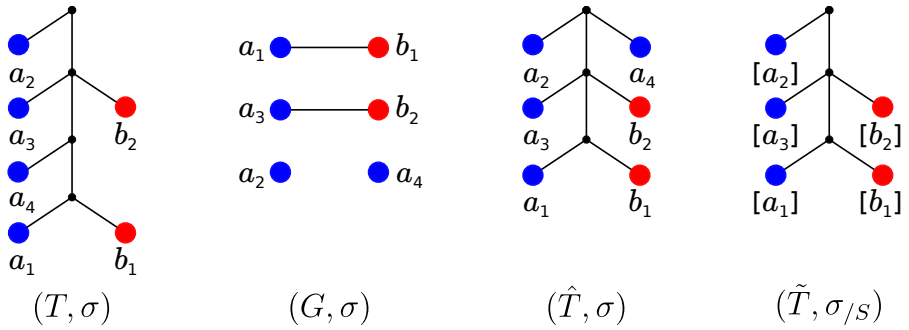


Fig. 20. The leaf-colored tree (T, σ) on the left explains the RBMG (G, σ), however $a_2, a_4 \in [a_2]$ but they do not have the same parent in T. The tree $(\hat{T}, \sigma)$ is obtained from (T, σ) by re-attaching the leaf a_2 to $\mathsf{par}(a_4)$ and suppressing the 2-degree vertex $\mathsf{par}(a_2)$. The resulting tree still explains (G, σ) and a_2 and a_4 are now siblings. Retaining only one representative of each S-class finally gives the tree $(\tilde{T}, \sigma_{/S})$ on the right that explains the S graph $(G_{/S}, \sigma_{/S})$.

by collapsing each class $[x]$ to a single leaf and potential suppression of 2-degree nodes still explains $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$.

The construction of T_{x',v_x} as in Lemma 5.5 can be repeated until all vertices x' of each S-class $[x]$ have been re-attached to have the same parent v_x. After each re-attachment step, the tree still explains (G, σ). The procedure stops when all $x' \in [x]$ are siblings of x in the tree, i.e., a tree $(\widehat{T}, \sigma)$ of the desired form is reached. The tree obtained by retaining only one representative of each S-class $[x]$ (relabeled as $[x]$), explains $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$.

Conversely, assume that $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$ is an RBMG explained by the tree $(\tilde{T}, \sigma_{/\mathsf{S}})$. Each leaf in $\tilde{T}$ is an S-class $[x]$. Consider the tree (T, σ) obtained by replacing, for all S-classes $[x]$ the edge $\mathsf{par}([x])[x]$ in T by the edges $\mathsf{par}([x])x'$ and setting $\sigma(x') = \sigma_{/\mathsf{S}}([x])$ for all $x' \in [x]$. By construction, (T, σ) explains (G, σ) and thus, (G, σ) is an RBMG. □

Lemma 5.6 is illustrated in Fig. 20, where the two leaves a_2 and a_4 belong to the same S-class $[a_2]$. However, in the tree representation on the l.h.s., a_2 and a_4 are attached to different parents. Substituting the edge $\mathsf{par}(a_4)a_4$ by $\mathsf{par}(a_2)a_4$ and suppressing the vertex $\mathsf{par}(a_4)$, which now has degree 2, yields a tree $(\hat{T}, \sigma)$ with $\mathsf{par}(a_2) = \mathsf{par}(a_4)$ that still explains (G, σ). Next, we can remove the edges $\mathsf{par}(a_2)a_2$ and $\mathsf{par}(a_2)a_4$ as well as the leaves a_2 and a_4 from $(\hat{T}, \sigma)$ and add the edge $\mathsf{par}(a_2)[a_2]$. Finally, we replace any vertex $y \neq a_2, a_4$ by $[y]$ and set $\sigma(x) = \sigma_{/\mathsf{S}}([x])$ for all $x \in V(\hat{T})$. The resulting tree explains the S-thin RBMG $(G_{/\mathsf{S}}, \sigma_{/\mathsf{S}})$.

Lemma 5.7. *Let (G, σ) be an S-thin n-RBMG explained by (T, σ) with $n \geq 2$. Then $|\sigma(L(T(v)))| \geq 2$ holds for every inner vertex $v \in V^0(T)$.*

Proof. Let $S = \sigma(V(G))$. Assume, for contradiction, that there exists an inner vertex $v \in V^0(T)$ such that $\sigma(L(T(v))) = \{r\}$ with $r \in S$. Since (T, σ) is phylogenetic, there must be two distinct leaves $a, b \in L(T(v))$ with $\sigma(a) = \sigma(b) = r$. Since (G, σ) is S-thin, a and b do not belong to the same S-class. Hence, $\sigma(a) = \sigma(b)$ implies $N(a) \neq N(b)$. Since $|S| \geq 2$, there is a leaf

$c \in V(G)$ with $\sigma(c) = s \in S \setminus \{r\}$. On the other hand, $\sigma(L(T(v))) = \{r\}$ implies $\mathrm{lca}(a, c) = \mathrm{lca}(b, c) \succ v$.

Now consider the corresponding BMG $\vec{G}(T, \sigma)$. Since $\sigma(L(T(v))) = \{r\}$, we have $c \in N^-(a)$ if and only if $c \in N^-(b)$, and $c \in N^+(a)$ if and only if $c \in N^+(b)$. Together this implies $N(a) = N(b)$ in $G(T, \sigma)$; a contradiction. □

Any two leaves x, y in (T, σ) with $\sigma(x) = \sigma(y)$ and $\mathsf{par}(x) = \mathsf{par}(y)$ obviously belong to the same S-equivalence class of $G(T, \sigma)$. The absence of such pairs of vertices in (T, σ) is thus a necessary condition for $G(T, \sigma)$ to be S-thin, it is not sufficient, however. The characterization of leaf-colored trees that explain S-thin RBMGs is left as an open question for future research.

5.4 Connected components, forks, and color-complete subtrees

This section aims at simplifying the problem of finding a characterization for RBMGs by showing that an undirected colored graph is an RBMG if and only if each of its connected components is an RBMG and one of those components contains all colors (cf. Theorem 5.3). This, in turn, allows us to consider connected graphs only. To this end, we will introduce so-called *forks* and *color-complete subtrees* and start by deriving some interesting and helpful results about those structures.

BMGs are not hereditary, hence we cannot expect RBMGs to be hereditary. They do satisfy a somewhat weaker property, however:

Lemma 5.8. *Let (G, σ) be an RBMG with vertex set L explained by (T, σ) and let $(T_{|L'}, \sigma_{|L'})$ be the restriction of (T, σ) to $L' \subseteq L$. Then the induced subgraph $(\vec{G}[L'], \sigma_{|L'})$ of $(\vec{G}, \sigma)$ is a (not necessarily induced) subgraph of $\vec{G}(T_{|L'}, \sigma_{|L'})$.*

Proof. Lemma 4.1 states the analogous result for BMGs. It obviously remains true for the symmetric part. □

The next result is a direct consequence of Lemma 5.8 that will be quite useful for proving some of the following results.

Corollary 5.2. *Let (G, σ) be an RBMG that is explained by (T, σ). Moreover, let $v \in V(T)$ be an arbitrary vertex and (G_v^*, σ_v^*) be a connected component of $G(T(v), \sigma_{|L(T(v))})$. Then (G_v^*, σ_v^*) is contained in a connected component (G^*, σ^*) of (G, σ).*

The following technical results ensures the existence of certain types of edges in any RBMG.

Lemma 5.9. *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 is an edge $xy \in E(G(T, \sigma))$ with $x \in L[r] \cap L(T(v))$ and $y \in L[s] \cap L(T(v))$. In particular, all edges in $G(T(v), \sigma_{|L(T(v))})$ are contained in $G(T, \sigma)$.*

Proof. Let v be a vertex of (T, σ) such that $r, s \in \sigma(L(T(v)))$, $r \neq s$. Then there is always an inner vertex $w \preceq_T v$ such that (i) $\{r, s\} \subseteq \sigma(L(T(w)))$ and

(ii) none of its children $w_i \in \mathsf{child}(w)$ satisfies $\{r, s\} \subseteq \sigma(L(T(w_i)))$. Any such w has children $w_r, w_s \in \mathsf{child}(w)$ such that $r \in \sigma(L(T(w_r)))$, $s \notin \sigma(L(T(w_r)))$ and $s \in \sigma(L(T(w_s)))$, $r \notin \sigma(L(T(w_s)))$. Thus $\mathrm{lca}_T(x, y) \succeq_T w$ for every $x \in L(T(w_r)) \cap L[r] \neq \emptyset$ and $y \in L[s]$, with equality whenever $y \in L(T(w_s))$. Analogously, $\mathrm{lca}_T(y, x) \succeq_T w$ for every $y \in L(T(w_s)) \cap L[s] \neq \emptyset$ and $x \in L[r]$, with equality whenever $x \in L(T(w_r))$. Hence, xy is a reciprocal best match mediated by $\mathrm{lca}_T(x, y) = w$ whenever $x \in L(T(w_r)) \cap L[r]$ and $y \in L(T(w_s)) \cap L[s]$. Therefore $xy \in E(G(T, \sigma))$.

In particular, the latter construction shows that the chosen leaves $x \in L(T(w_r)) \cap L[r]$ and $y \in L(T(w_s)) \cap L[s]$ are reciprocal best matches in $(T(v), \sigma_{|L(T(v))})$. Hence, every edge in $G(T(v), \sigma_{|L(T(v))})$ is also contained in $G(T, \sigma)$. □

As a direct consequence of Lemma 5.9, we obtain

Corollary 5.3. *If (G, σ) is an RBMG with $|S| \geq 2$ colors, then there is at least one edge $xy \in E(G[L[r] \cup L[s]])$ for any two distinct colors $r, s \in S$.*

As noted above, the property of being an RBMG is not hereditary. Thm. 5.2 below shows that the connected components of an RBMG are again RBMGs that can be explained by corresponding restrictions of a leaf-colored tree, although there is no similar result for BMGs.

Theorem 5.2. *Let (G^*, σ^*) with vertex set L^* be a connected component of some RBMG (G, σ) and let (T, σ) be a leaf-colored tree explaining (G, σ). Then (G^*, σ^*) is again an RBMG and is explained by the restriction $(T_{|L^*}, \sigma_{|L^*})$ of (T, σ) to L^*.*

Proof. Throughout this proof, all N^+-neighborhoods are taken w.r.t. the underlying BMG $\vec{G}(T, \sigma)$. It suffices to show that $G(T_{|L^*}, \sigma_{|L^*}) = (G^*, \sigma^*)$. Lemma 5.8 implies that (G^*, σ^*) is a (not necessarily induced) subgraph of $G(T_{|L^*}, \sigma_{|L^*})$, i.e., $E(G^*) \subseteq E(G(T_{|L^*}, \sigma_{|L^*}))$. By assumption, (G^*, σ^*) is an induced subgraph of (G, σ). Thus we only need to prove $E(G(T_{|L^*}, \sigma_{|L^*})) \subseteq E(G^*)$.

Assume, for contradiction, that there exists an edge xy in $G(T_{|L^*}, \sigma_{|L^*})$ that is not contained in (G^*, σ^*). By definition, $r := \sigma(x) \neq s := \sigma(y)$ and, in particular, $x, y \in L^*$. Let $u := \mathrm{lca}_T(x, y)$. By construction, any two vertices within L^* have the same last common ancestor in (T, σ) and $(T_{|L^*}, \sigma_{|L^*})$. Since the edge xy is not contained in (G^*, σ^*), the edge xy is not contained in (G, σ) either. Hence, x and y do not form reciprocal best matches in (T, σ). Thus there must exist some $x' \in L[r]$ with $\mathrm{lca}_T(x', y) \prec_T \mathrm{lca}_T(x, y)$, or a leaf $y' \in L[s]$ with $\mathrm{lca}_T(x, y') \prec_T \mathrm{lca}_T(x, y)$.

W.l.o.g. we assume that the first case is satisfied. Since $\mathrm{lca}_T(x', y) \prec_T \mathrm{lca}_T(x, y)$, we must have $x' \in L \setminus L^*$, as otherwise, $\mathrm{lca}_{T_{|L^*}}(x', y) \prec_{T_{|L^*}} \mathrm{lca}_{T_{|L^*}}(x, y)$ and hence, x cannot be a best match of y, which in turn would imply that xy is not an edge in $G(T_{|L^*}, \sigma_{|L^*})$. We will re-use the latter argument and refer to it as *Argument-1*.

In the following, w.l.o.g. we choose $x' \in L[r]$ such that $\mathrm{lca}_T(x', y) \prec_T \mathrm{lca}_T(x, y)$ and $\mathrm{lca}(x', y)$ is $\preceq_T$-minimal among all least common ancestors that

satisfy the latter condition. We write $v := \mathrm{lca}_T(x', y)$. By construction, we have $v \prec_T u$. By contraposition of *Argument-1*, it must hold for all $x'' \in L^*$ with $\sigma(x'') = r$ that $\mathrm{lca}_T(x'', y) \succeq_T \mathrm{lca}_T(x, y)$ and thus, $x'' \notin L(T(v))$. In other words, we have

$$x'' \notin L^* \text{ for all } x'' \in L(T(v)) \cap L[r]. \tag{15}$$

Let $v_{x'}, v_y \in \mathsf{child}(v)$ with $x' \preceq_T v_{x'}$ and $y \preceq_T v_y$. The choice of x' and the resulting $\preceq_T$-minimality of $\mathrm{lca}(x', y)$ implies that $\sigma(x) = r \notin \sigma(L(T(v_y)))$. Therefore $x' \in N_r^+(y)$. We observe that $x'y \notin E(G)$ since $x' \in L^*$ otherwise; a contradiction. From $x'y \notin E(G)$ we conclude $y \notin N_s^+(x')$ and thus, there exists a leaf $y' \in L[s]$ such that $\mathrm{lca}_T(x', y') \prec_T \mathrm{lca}_T(x', y) = v$ and hence, $y' \prec_T v_{x'}$.

The latter, in particular, implies $r, s \in \sigma(L(T(v_{x'})))$. Hence, we can apply Lemma 5.9 to conclude that there are two vertices $\tilde{x} \in L[r] \cap L(T(v_{x'}))$ and $\tilde{y} \in L[s] \cap L(T(v_{x'}))$ such that $\tilde{x}\tilde{y} \in E(G)$. Equ. (15) now implies $\tilde{x} \notin L^*$. Therefore $\tilde{x}\tilde{y} \in E(G)$ now allows us to conclude that $\tilde{y} \in L \setminus L^*$.

Now, let $\mathcal{P}_{xy} = (x = a_0 a_1 a_2 \dots a_{k-1} a_k = y)$ be a shortest path in (G^*, σ^*) connecting x and y. Since x and y reside within the same connected component (G^*, σ^*) of (G, σ) and $xy \notin E(G^*)$, such a path exists and, in particular, it must contain at least one $a_i \neq x, y$, i.e., $k > 1$. By definition of a shortest path, $a_i a_j \notin E(G)$ for all $i, j \in \{0, 1, \dots, k\}$ that satisfy $|i - j| > 1$. Since $a_i \in L^*$ for any $0 \leq i \leq k$ but $\tilde{x}, \tilde{y} \in L \setminus L^*$, we have

$$\tilde{x}a_i, \tilde{y}a_i \notin E(G) \tag{16}$$

for any $0 \leq i \leq k$, since otherwise, $\tilde{x}$ and $\tilde{y}$ would be contained in the connected component (G^*, σ^*) and thus, also in L^*; a contradiction.

We proceed to show by induction that

(I1) $a_i \in L(T(v))$, $1 \leq i \leq k$, and

(I2) there exists a vertex $\tilde{a}_i \in L(T(v)) \cap L[\sigma(a_i)]$ such that $\tilde{a}_i \notin L^*$, $1 \leq i \leq k$.

We start with $i = k$. By construction, $y = a_k \in L(T(v))$ satisfies Property (I1). Moreover, $\tilde{a}_k := \tilde{y}$ satisfies Property (I2). For the induction step assume that, for a fixed $m \leq k$, Property (I1) and (I2) is satisfied for all i with $m < i \leq k$.

Now consider the case $i = m$. For better readability we put $b := a_{m+1}$ and $\tilde{b} := \tilde{a}_{m+1}$. By induction hypothesis, b and $\tilde{b}$ satisfy Property (I1) and (I2), respectively. Since $a_m b \in E(G)$, we know that $\sigma(a_m) \neq \sigma(b)$. In the following, we consider the two exclusive cases: either $\sigma(a_m) = \sigma(x) = r$ or $\sigma(a_m) \neq r$. If $\sigma(a_m) = r$, then we put $\tilde{a}_m = \tilde{x}$. Hence, Property (I2) is trivially satisfied for $\tilde{a}_m$. Moreover, a_m must then be contained in $L(T(v))$, otherwise $v \succeq_T \mathrm{lca}_T(b, \tilde{x})$ implies that $\mathrm{lca}_T(b, \tilde{x}) \prec_T \mathrm{lca}(b, a_m)$, which contradicts $a_m b \in E(G)$, i.e., Property (I1) is satisfied as well.

In case $\sigma(a_m) \neq r$ assume first, for contradiction, that $a_m \notin L(T(v))$. Since $b, \tilde{b} \in L(T(v))$ we observe that $\mathrm{lca}_T(b, a_m) = \mathrm{lca}_T(\tilde{b}, a_m) \succ_T v$. Note that we have $b \in N^+_{\sigma(b)}(a_m)$ since $ba_m \in E(G)$ by definition of $\mathcal{P}_{xy}$. Thus $\mathrm{lca}_T(b, a_m) = \mathrm{lca}_T(\tilde{b}, a_m)$ implies $\tilde{b} \in N^+_{\sigma(b)}(a_m)$. Since $a_m \in L^*$ (by definition) and $\tilde{b} \notin L^*$ (by Property (I2)), we can conclude that $a_m\tilde{b} \notin E(G)$. The latter two

arguments imply $a_m \notin N^+_{\sigma(a_m)}(\tilde{b})$. Hence, there exists a leaf a'_m with $\sigma(a_m) = \sigma(a'_m)$ such that $\mathrm{lca}_T(\tilde{b}, a'_m) \prec_T \mathrm{lca}_T(\tilde{b}, a_m)$. There are two cases, either $a'_m \in L(T(v_{\tilde{b}}))$ or $a'_m \notin L(T(v_{\tilde{b}}))$, where $v_{\tilde{b}} \in \mathsf{child}(v)$ with $\tilde{b} \preceq_T v_{\tilde{b}}$. If $a'_m \in L(T(v_{\tilde{b}}))$, then $\mathrm{lca}_T(b, a'_m) \preceq_T v$ and we can re-use the fact $\mathrm{lca}_T(b, a_m) \succ_T v$ from above to conclude that $\mathrm{lca}_T(b, a'_m) \prec_T \mathrm{lca}_T(b, a_m)$. If $a'_m \notin L(T(v_{\tilde{b}}))$, then $\mathrm{lca}_T(b, a'_m) \preceq_T \mathrm{lca}_T(\tilde{b}, a'_m)$. Thus we have $\mathrm{lca}_T(b, a'_m) \preceq_T \mathrm{lca}_T(\tilde{b}, a'_m) \prec_T \mathrm{lca}_T(\tilde{b}, a_m) = \mathrm{lca}_T(b, a_m)$. Hence, in either case we obtain $\mathrm{lca}_T(b, a'_m) \prec_T \mathrm{lca}_T(b, a_m)$, thus $a_m b \notin E(G)$; a contradiction. Therefore $a_m \in L(T(v))$, i.e., Property (I1) is satisfied by a_m.

To summarize the argument so far, Property (I1) is always satisfied for a_m, independent of the particular color $\sigma(a_m)$. Moreover, Property (I2) is satisfied, in case $\sigma(a_m) = r$. Thus it remains to show that Property (I2) is also satisfied in case $\sigma(a_m) \neq r$. To this end, let $v_m \in \mathsf{child}(v)$ such that $a_m \preceq_T v_m$. If $r \in \sigma(L(T(v_m)))$, then Lemma 5.9 implies that there must exist leaves $\tilde{x}_m, \tilde{a}_m \in L(T(v_m))$ with $\sigma(\tilde{x}_m) = r$ and $\sigma(\tilde{a}_m) = \sigma(a_m)$ such that $\tilde{x}_m \tilde{a}_m \in E(G)$. By Equ. (15), no vertex in $L(T(v)) \cap L[r]$ is contained in L^*, and thus, we have $\tilde{x}_m \notin L^*$ and, since $\tilde{x}_m \tilde{a}_m \in E(G)$, it must also hold $\tilde{a}_m \notin L^*$.

Otherwise, if $r \notin \sigma(L(T(v_m)))$, then $\sigma(\tilde{x}) = r$ and $\tilde{x} \preceq_T v_{x'}$ implies $v_m \neq v_{x'}$. Hence, $\mathrm{lca}(a_m, \tilde{x}) = v$. In particular, there is no vertex $x'' \in L[r]$ such that $\mathrm{lca}_T(a_m, x'') \prec_T \mathrm{lca}(a_m, \tilde{x}) = v$, thus $\tilde{x} \in N^+_r(a_m)$. Since $a_m \in L^*$ and $\tilde{x} \notin L^*$, it must hold $a_m \tilde{x} \notin E(G)$. Thus there must exist a leaf $\tilde{a}_m \in L[\sigma(a_m)]$ such that $\mathrm{lca}_T(\tilde{x}, \tilde{a}_m) \prec_T \mathrm{lca}_T(\tilde{x}, a_m) = v$, i.e., $\sigma(a_m) \in \sigma(L(T(v_{x'})))$. We can therefore apply Lemma 5.9 to conclude that there must exist $\tilde{x}_m \in L(T(v_{x'})) \cap L[r]$ and $\tilde{a}_m \in L(T(v_{x'})) \cap L[\sigma(a_m)]$ such that $\tilde{x}_m \tilde{a}_m \in E(G)$. Analogous argumentation as in the case $r \in \sigma(L(T(v_m)))$ shows $\tilde{x}_m, \tilde{a}_m \notin L^*$. Hence, Property (I2) is satisfied, which completes the induction proof.

Property (I1) finally implies that $a_1 \in L(T(v))$. Moreover, by construction of $\mathcal{P}_{xy}$ we have $x a_1 \in E(G^*)$. Property (16), on the other hand, implies $\tilde{x} a_1 \notin E(G^*)$. Consequently, we have $\mathrm{lca}_T(a_1, x) \prec_T \mathrm{lca}_T(a_1, \tilde{x})$. This, however, contradicts $\mathrm{lca}_T(a_1, x) = u \succ_T v = \mathrm{lca}_T(a_1, \tilde{x})$. The shortest path $\mathcal{P}_{xy}$ can, therefore, consist only of the single edge xy and hence, $E(G(T_{|L^*}, \sigma_{|L^*})) \subseteq E(G^*)$. Therefore $G(T_{|L^*}, \sigma_{|L^*}) = (G^*, \sigma^*)$ and $(T_{|L^*}, \sigma_{|L^*})$ explains (G^*, σ^*). In particular, the connected component (G^*, σ^*) is again an RBMG.

□

Every connected component of an n-RBMG is therefore a k-RBMG possibly with a strictly smaller number k of colors. Our aim in the remainder of this section is to show that the disjoint union of RBMGs is again an RBMG. We start by identifying certain vertices in the leaf-colored tree (T, σ) that, as we shall see below, are related to the decomposition of $G(T, \sigma)$ into connected components.

Definition 5.6. *Let (T, σ) be a leaf-colored tree with leaf set L. An inner vertex u of T is* color-complete *if $\sigma(L(T(u))) = \sigma(L)$. Otherwise, it is* color-deficient.

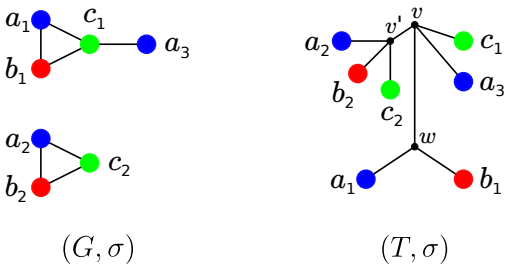


Fig. 21. The 3-RBMG (G, σ) on the left hand side can be explained by the tree (T, σ) shown on the right. In (T, σ), the inner vertex v is a fork. The color-deficient children of v are c_1, a_3 and w, thus $\mathcal{L}(v) = \{a_1, b_1, c_1, a_3\}$. Also, v' is a fork and $\mathcal{L}(v') = \{a_2, b_2, c_2\}$. The set of forks is $\zeta(T, \sigma) = \{v, v'\}$.

We will also refer to a subtree $(T(u), \sigma_{|L(T(u))})$ of (T, σ) as *color-complete* if its root is color-complete.
We write $A(u)$ for the set of color-deficient children of u, i.e.,

$$A(u) := \{v \mid v \in \mathsf{child}(u), \sigma(L(T(v))) \subsetneq \sigma(L)\} \tag{17}$$

and set

$$\mathcal{L}(u) := \bigcup_{v \in A(u)} L(T(v)). \tag{18}$$

Definition 5.7. *Let (T, σ) be a leaf-colored tree. An inner vertex $u \in V^0(T)$ is a* fork *if $\sigma(\mathcal{L}(u)) = \sigma(L)$. We write $\zeta(T, \sigma)$ for the set of forks in (T, σ).*

For an illustration see Fig. 21. As an immediate consequence of the definition we have

Lemma 5.10. *Every fork in a leaf-colored tree (T, σ) is color-complete, but not every color-complete vertex is a fork.*

Proof. For a fork u, we have $\sigma(L(T)) = \sigma(\mathcal{L}(u)) \subseteq \bigcup_{v \in \mathsf{child}(u)} \sigma(L(T(v))) = \sigma(L(T(u)))$. Thus every fork must be color-complete. In order to see that not every color-complete vertex is a fork, consider a leaf-colored tree (T, σ), where ρ_T has exactly two children both of which are color-complete. Then ρ_T is color-complete but $A(\rho_T) = \emptyset$. Hence, ρ_T is not a fork. □

Clearly, there are no forks in a leaf-labeled tree (T, σ) with $|\sigma(L(T))| = 1$. We will therefore in the following restrict our attention to trees and graphs with at least two colors, omitting the trivial case of 1-RBMGs which correspond to the edge-less graph and are explained by any leaf-colored tree with the same leaf set. Next, we derive some useful technical results about forks and color-complete trees, which will be needed to prove the main result of this section.

Lemma 5.11. *Let (T, σ) be a leaf-colored tree. Then $\zeta(T, \sigma) \neq \emptyset$.*

Proof. Let $L = L(T)$. Assume, for contradiction, that $\zeta(T, \sigma) = \emptyset$. Thus, in particular, $\rho_T \notin \zeta(T, \sigma)$. Since the root ρ_T is always color-complete, we have $\sigma(\mathcal{L}(\rho_T)) \neq \sigma(L(T(\rho_T))) = \sigma(L)$, which implies $A(\rho_T) \subsetneq \mathsf{child}(\rho_T)$. Hence, Equ. (17) implies that there is a child u_1 of the root with $\sigma(L(T(u_1))) = \sigma(L)$.

Since $\zeta(T,\sigma) = \emptyset$, the vertex u_1 is not a fork. Repeating the argument, u_1 must have a child u_2 with $\sigma(L(T(u_2))) = \sigma(L)$, and so on. Hence, there is a sequence of inner vertices $\rho_T := u_0 \succ_T u_1 \succ_T u_2 \succ_T \cdots \succ_T u_k$ such that u_j has only color-complete children for $0 \leq j < k$. Since T is finite, all maximal paths of this form a finite, i.e., the final vertex u_k in every maximal path has only color-deficient children, i.e., $A(u) = \mathsf{child}(u)$. Since u_k itself is color-complete by construction, $\sigma(\mathcal{L}(u)) = \sigma(L(T(u))) = \sigma(L)$, i.e., u_k is fork, a contradiction.

□

Lemma 5.12. *Let (G,σ) be an n-RBMG, $n \geq 2$, (T,σ) a tree with leaf set L that explains (G,σ), and $(T(u),\sigma_{|L(T(u))})$ a color-complete subtree of (T,σ) for some $u \in V^0(T)$. Then $xy \notin E(G)$ for any two vertices $x,y \in L$ with $x \in L(T(u))$ and $y \in L \setminus L(T(u))$.*

Proof. If $u = \rho_T$, then $L \setminus L(T(u)) = \emptyset$ and the lemma is trivially true. Thus suppose $u \neq \rho_T$. Let $x \in L(T(u))$ and assume, for contradiction, $xy \in E(G)$ for some $y \in L \setminus L(T(u))$, i.e., x and y are reciprocal best matches. By choice of x and y, $\mathrm{lca}(x,y) \succ u$ and $\sigma(x) \neq \sigma(y)$. Since $(T(u),\sigma_{|L(T(u))})$ is color-complete, there exists a leaf $y' \in L(T(u))$ with $\sigma(y') = \sigma(y)$. Hence, in particular, $\sigma(y') \neq \sigma(x)$ and thus, $y' \neq x$. Since $y' \in L(T(u))$, we have $\mathrm{lca}(x,y') \preceq u \prec \mathrm{lca}(x,y)$; a contradiction to the assumption that x and y are reciprocal best matches.

□

Lemma 5.13. *Let (T,σ) be a leaf-colored tree with leaf set L that explains the n-RBMG (G,σ), and let $u \in \zeta(T,\sigma)$ be a fork in (T,σ). Then the following statements are true:*

(i) If L^* is the vertex set of a connected component (G^*,σ^*) of (G,σ), then either $L^* \subseteq \mathcal{L}(u)$ or $L^* \cap \mathcal{L}(u) = \emptyset$.

(ii) If $n \geq 2$, then there is a connected component (G^*,σ^*) of (G,σ) with leaf set $L^* \subseteq \mathcal{L}(u)$ and $\sigma(L^*) = \sigma(L)$.

(iii) Let (G^*,σ^*) be a connected component of (G,σ) with vertex set L^* and $\sigma(L^*) = \sigma(L)$. If $n \geq 2$, then $u' := \mathrm{lca}(L^*)$ is a fork and $L^* \subseteq \mathcal{L}(u')$.

Proof. All N^+-neighborhoods in this proof are taken w.r.t. the underlying BMG $\vec{G}(T,\sigma)$. By Lemma 5.11, we have $\zeta(T,\sigma) \neq \emptyset$ and thus, there exists a fork in (T,σ). In the following, let $u \in \zeta(T,\sigma)$ be chosen arbitrarily.

(i) Let (G^*,σ^*) be a connected component of (G,σ) and L^* its vertex set. The statement is trivially true if $|L^*| = 1$. Hence, assume $|L^*| \geq 2$. By Lemma 5.10, $(T(u),\sigma_{|L(T(u))})$ is color-complete. Lemma 5.12 implies $xy \notin E(G)$ for any pair of leaves $x \in L(T(u))$ and $y \in L \setminus L(T(u))$. Therefore either $L^* \subseteq L(T(u))$ or $L^* \cap L(T(u)) = \emptyset$. In the latter case, we have $L^* \cap \mathcal{L}(u) \subseteq L^* \cap L(T(u)) = \emptyset$. Now suppose $L^* \subseteq L(T(u))$. Consider a vertex $x \in L^*$ and let $z \in L^* \setminus \{x\}$ be a neighbor of x, i.e., $xz \in E(G)$. Such a vertex z exists since $|L^*| \geq 2$ and G^* is connected. If $x \in L(T(u)) \setminus \mathcal{L}(u)$, then there exists a color-complete inner vertex $v \in \mathsf{child}(u)$ that satisfies $x \prec v$. Since v is color-complete, Lemma 5.12 implies that there is no edge between $L(T(v))$ and $L(T(u)) \setminus L(T(v))$ and

thus, we have $z \in L(T(v))$. Therefore $z \notin \mathcal{L}(u)$. Now suppose $x \in \mathcal{L}(u)$. If $z \notin \mathcal{L}(u)$, then $z \in L(T(u)) \setminus \mathcal{L}(u)$. Thus we can apply analogous arguments and Lemma 5.12 to conclude that there cannot be an edge between x and z; a contradiction. Hence, $z \in \mathcal{L}(u)$. In summary, we have either $L^* \subseteq \mathcal{L}(u)$ or $L^* \cap \mathcal{L}(u) = \emptyset$.

(ii) Let $S := \sigma(L)$ with $|S| = n > 1$. We proceed by induction. For $n = 2$, the statement is a direct consequence of Lemma 5.9.

For the induction step, suppose the statement is correct for RBMGs with a color set of less than n colors. Recall that for any $v_i \in A(u)$ the color set of any subtree $(T(v_i), \sigma_{|L(T(v_i))})$ contains less than n colors, i.e., $S_{v_i} := \sigma(L(T(v_i))) \neq S$. By Lemma 5.11, there must exist a fork $w \in \zeta(T(v_i), \sigma_{|L(T(v_i))})$ within the tree $(T(v_i), \sigma_{|L(T(v_i))})$. Since w is a fork in $(T(v_i), \sigma_{|L(T(v_i))})$, it is therefore also color-complete in $(T(v_i), \sigma_{|L(T(v_i))})$. However, by definition, we have $w \preceq v_i \in A(u)$ and thus, w is not color-complete in (T, σ). Nevertheless, we can apply the induction hypothesis to the RBMG $(G_{v_i}, \sigma_{v_i}) := G(T(v_i), \sigma_{|L(T(v_i))})$ to ensure that there exists a connected component $(G^*_{v_i}, \sigma^*_{v_i})$ with leaf set $L^*_{v_i} \subseteq \mathcal{L}(w)$ and $\sigma(L^*_{v_i}) = S_{v_i}$. Now fix this index i. By Cor. 5.2, there is a connected component (G^*, σ^*) with leaf set L^* of (G, σ) that contains $(G^*_{v_i}, \sigma^*_{v_i})$.

Assume for contradiction $|\sigma(L^*)| < n$. Suppose first $|S_{v_i}| = n - 1$. Thus $S \setminus S_{v_i} = \{r\}$ and for each color $s \in S \setminus \{r\}$ there is a vertex $z \in V(G^*_{v_i})$ with color $\sigma(z) = s$. By construction, $u \in \zeta(T, \sigma)$ implies that there exists a vertex $v_j \in A(u)$ $(i \neq j)$ such that $r \in S_{v_j}$. In particular, it follows from Lemma 4.14 that $L(T(v_j)) \cap L[r] \subseteq N^+_r(x)$ for all $x \in L(T(v_i))$. Since $S_{v_i} \subseteq \sigma(L^*)$ but $|\sigma(L^*)| < n$, we have $|\sigma(L^*)| = n - 1$, and we conclude that $xy \notin E(G)$ for every $y \in L(T(v_j)) \cap L[r]$ and $x \in V(G^*_{v_i})$. The latter two arguments imply that $x \notin N^+_{\sigma(x)}(y)$ for all $y \in L(T(v_j)) \cap L[r]$ and $x \in V(G^*_{v_i})$. This, however, is only possible if $L(T(v_j))$ contains leaves of all colors $s \neq r$, i.e., $S_{v_i} \subsetneq S_{v_j}$ and thus, $|S_{v_j}| = n$; a contradiction to $v_j \in A(u)$.

Now suppose $|S_{v_i}| < n - 1$, i.e., $S \setminus S_{v_i} = \{r_1, ..., r_m\}$. Again, for any r_j $(1 \leq j \leq m)$, there is a vertex $v_j \in A(u)$ $(i \neq j)$ such that $r_j \in S_{v_j}$. Note that $v_j = v_k$ may be possible for two different colors r_j and r_k. If there exists a color $s_j \in S_{v_i}$ that is not contained in S_{v_j}, then, for any $x \in L(T(v_j)) \cap L[r_j]$ and $y \in L(T(v_i)) \cap L[s_j]$, we have $\mathrm{lca}_T(x, y) = u \prec_T \mathrm{lca}_T(x, y')$ and $\mathrm{lca}_T(x, y) = u \prec_T \mathrm{lca}_T(x', y)$ for all $x' \in L[r_j], y' \in L[s_j]$ and hence, $xy \in E(G)$. Thus there is a connected component in $G(T(u), \sigma_{|L(T(u))})$ that contains at least all colors $S_{v_i} \cup \{r_j\}$. Consequently, if for any $j \in \{1, \ldots, m\}$ there exists such a color $s_j \in S_{v_i} \setminus S_{v_j}$, then there must be a connected component in $G(T(u), \sigma_{|L(T(u))})$ that contains all colors in S. By Cor. 5.2, every connected component of $G(T(u), \sigma_{|L(T(u))})$ is contained in a connected component of (G, σ) and statement (ii) is true for this case.

On the other hand, if there is at least one j for which this property is not true, similar argumentation as in the case $|S_{v_i}| = n - 1$ shows that $S_{v_i} \subset S_{v_j}$, hence in particular $|S_{v_j}| > |S_{v_i}|$. We can then apply the same argumentation to the RBMG $(G_{v_j}, \sigma_{v_j}) := G(T(v_j), \sigma_{|L(T(v_j))})$ and either obtain a connected component on n colors in $G(T(u), \sigma_{|L(T(u))})$ or some inner vertex $v_k \in A(u)$ with $|S_{v_i}| < |S_{v_j}| < |S_{v_k}|$. Repeating this argumentation, in each step we either obtain an n-colored connected component or further increase the sequence

$|S_{v_i}| < |S_{v_j}| < |S_{v_k}| < \dots$. Since L is finite, this sequence must eventually terminate with $|S_{v_l}| = n$, contradicting $v_l \in A(u)$. In summary, we have shown that $|\sigma(L^*)| \neq n$ is not possible and hence, $\sigma(L^*) = \sigma(L)$. Finally, $\emptyset \neq L^* \cap L(T(v_i))$ and $v_i \in A(u)$ implies $L^* \cap \mathcal{L}(u) \neq \emptyset$. Thus we can apply Statement (i) to conclude that $L^* \subseteq \mathcal{L}(u)$.

(iii) By Statement (ii), there is a connected component (G^*, σ^*) with vertex set L^* and $\sigma(L^*) = \sigma(L)$. Put $u' := \mathrm{lca}(L^*)$. We start by showing $L^* \subseteq \mathcal{L}(u')$. Assume, for contradiction, that there exists a leaf $a \in L^*$ such that $a \notin \mathcal{L}(u')$. Let $v' \in \mathsf{child}(u')$ be the (unique) child of u' with $a \preceq_T v'$. Since $a \notin \mathcal{L}(u')$, we can conclude that $v' \notin A(u')$. Thus v' is color-complete and therefore, $(T(v'), \sigma_{|L(T(v'))})$ is color-complete. By Lemma 5.12, we thus have $b \prec_T v'$ for any $b \in L$ with $ab \in E(G)$. Repeating this argument for any $b \in N(a)$ and $c \in N(b)$ and so on, this finally implies $L^* \subseteq_T L(T(v'))$. Therefore $\mathrm{lca}(L^*) \preceq_T v' \prec_T u'$; a contradiction to $u' = \mathrm{lca}(L^*)$. Thus we have $L^* \subseteq \mathcal{L}(u')$. As a consequence, $\sigma(\mathcal{L}(u')) = \sigma(L)$, i.e., u' is a fork. □

Corollary 5.4. *Let (G, σ) be an n-RBMG, $n \geq 2$, that is explained by a tree (T, σ) with root ρ_T.*

(i) There exists an n-colored connected component (G^*, σ^*) of (G, σ).

(ii) If (G, σ) is connected, then $\zeta(T, \sigma) = \{\rho_T\}$.

Proof. (i) Since $\zeta(T, \sigma) \neq \emptyset$ (see Lemma 5.11), the existence of an n-colored connected component of (G, σ) is a direct consequence of Lemma 5.13(ii).
(ii) Lemma 5.13(iii) implies $\rho_T \in \zeta(T, \sigma)$. By (i) and Lemma 5.13(ii), we have $L(T) \subseteq \mathcal{L}(u) \subseteq L(T)$ for all $u \in \zeta(T, \sigma)$, hence $\mathcal{L}(u) = L(T)$. Since this is true only if $u = \rho_T$, Assertion (ii) follows. □

The following result helps to gain some understanding of the ambiguities among the leaf-colored trees that explain the same RBMG.

Lemma 5.14. *Let (G, σ) be an n-RBMG, $n \geq 2$, explained by (T, σ) and $u \in \zeta(T, \sigma)$ with $u \neq \rho_T$. Moreover, let $v \in \mathsf{child}(u)$, where v is color-complete, and (T', σ) the tree obtained from (T, σ) by replacing the edge uv by $\mathsf{par}(u)v$. Then (T', σ) explains (G, σ).*

Proof. First note that, since u is a fork in (T, σ), there must exist at least two color-deficient nodes $w_1, w_2 \in A(u)$. Since v is color-complete, we have $v \neq w_1, w_2$, thus $\deg_{T'}(u) > 2$, i.e., (T', σ) is phylogenetic. We first show $(G, \sigma) = G(T', \sigma)$. Put $L := V(G)$.

First, let $x, y \in L \setminus L(T(u))$. Then, by construction of (T', σ), we have $\mathrm{lca}_T(x, y) = \mathrm{lca}_{T'}(x, y)$, and $\mathrm{lca}_T(x, y) \prec_T z$ implies $\mathrm{lca}_{T'}(x, y) \prec_{T'} z$ for all $z \in V(T)$. In other words, reciprocal best matches xy with $x, y \notin L(T(u))$ remain reciprocal best matches in (T', σ). Moreover, if x and y are not reciprocal best matches in (T, σ), then we have w.l.o.g. $\mathrm{lca}_T(x, y) \succ_T \mathrm{lca}_T(x', y)$ for some (fixed) $x' \in L[\sigma(x)]$. Clearly, if $x' \in L \setminus L(T(v))$, then we still have, by construction, $\mathrm{lca}_{T'}(x, y) = \mathrm{lca}_T(x, y) \succ_{T'} \mathrm{lca}_{T'}(x', y) = \mathrm{lca}_T(x', y)$. Thus, if $x' \in L \setminus L(T(v))$, then x and y do not form reciprocal best matches in (T', σ). If $x' \in L(T(v))$, then $\mathrm{lca}_T(x', y) \succeq_T \mathsf{par}(u)$. Now, $\mathrm{lca}_T(x, y) \succ_T \mathrm{lca}_T(x', y)$

implies $\mathrm{lca}_T(x,y) \succ_T \mathsf{par}(u)$. In other words, $\mathrm{lca}_T(x',y)$ and $\mathrm{lca}_T(x,y)$ lie on the path from the root to $\mathsf{par}(u)$. This and the construction of (T',σ) implies $\mathrm{lca}_T(x,y) = \mathrm{lca}_{T'}(x,y) \succ_{T'} \mathrm{lca}_T(x',y) = \mathrm{lca}_{T'}(x',y)$. Thus x and y do not form reciprocal best matches in (T',σ). In summary, $xy \in E(G)$ if and only if $xy \in E(G(T,\sigma))$ for all $x,y \in L \setminus L(T(u))$.

Moreover, since v is color-complete in both trees, we can apply Lemma 5.12 to conclude that neither (G,σ) nor $G(T',\sigma)$ contains edges between $L(T(v))$ and $L \setminus L(T(v))$. Since $T'(v) = T(v)$ by construction, we additionally have $G(T'(v),\sigma_{|L(T(v))}) = G(T(v),\sigma_{|L(T(v))}) = (G[L(T(v))],\sigma_{|L(T(v))})$.

It remains to show the case $x \in L' := L(T(u)) \setminus L(T(v))$, and either $y \in L'$ or $y \in L \setminus L(T(u))$. Suppose first $y \in L \setminus L(T(u))$. Since u is a fork, Lemma 5.13(ii) implies that there exists a connected component (G^*,σ^*) of (G,σ) with leaf set L^* such that $L^* \subseteq \mathcal{L}(u)$. In particular, as v is color-complete, it is not contained in $A(u)$. We therefore conclude that $L^* \subseteq L'$, i.e., the subtree $(T_{|L'},\sigma_{|L'})$ is color-complete as well. Since, by construction, $T'(u) = T_{|L'}$, Lemma 5.12 implies that there are no edges between $L(T(u))$ and $L \setminus L(T(u))$ in both (G,σ) and $G(T',\sigma)$. In other words, x and y do not form reciprocal best matches, neither in (T,σ) nor in (T',σ) whenever $x \in L' := L(T(u)) \setminus L(T(v))$ and $y \in L \setminus L(T(u))$.

Now suppose $y \in L'$. If x and y do not form reciprocal best matches in (T,σ), then we have w.l.o.g. $\mathrm{lca}_T(x,y) \succ_T \mathrm{lca}_T(x',y)$ for some (fixed) $x' \in L[\sigma(x)]$. This immediately implies that $x' \in L'$. Again, since $T'(u) = T_{|L'}$, we have $\mathrm{lca}_T(x,y) = \mathrm{lca}_{T'}(x,y) \succ_{T'} \mathrm{lca}_T(x',y) = \mathrm{lca}_{T'}(x',y)$. Hence, x and y do not form reciprocal best matches in x and y in (T',σ). Finally, if x and y are reciprocal best matches in (T,σ), then $\mathrm{lca}_T(x,y) \preceq_T \mathrm{lca}_T(x',y)$ and $\mathrm{lca}_T(x,y) \preceq_T \mathrm{lca}_T(x,y')$ for all $x' \in L[\sigma(x)]$ and $y' \in L[\sigma(y)]$. We first fix a leaf $x' \in L[\sigma(x)]$ for which the latter inequality is satisfied. By construction, $\mathrm{lca}_T(x,y) = \mathrm{lca}_{T'}(x,y) \preceq_{T'} u$. Clearly, if $x' \in L'$, then the fact $T'(u) = T_{|L'}$ implies that $\mathrm{lca}_{T'}(x,y) \preceq_{T'} \mathrm{lca}_{T'}(x',y)$. On the other hand, if $x' \notin L'$, then $\mathrm{lca}_{T'}(x',y) \succ_{T'} u$ by construction of (T',σ). We thus have $\mathrm{lca}_{T'}(x,y) \preceq_{T'} u \prec_{T'} \mathrm{lca}_{T'}(x',y)$. Hence, $\mathrm{lca}_T(x,y) \preceq_T \mathrm{lca}_T(x',y)$ implies $\mathrm{lca}_{T'}(x,y) \preceq_{T'} \mathrm{lca}_{T'}(x',y)$ for all $x' \in L[\sigma(x)]$. Analogous arguments hold for $y' \in L[\sigma(y)]$. Hence, x and y remain reciprocal best matches in (T',σ).
In summary, $xy \in E(G)$ if and only $xy \in E(G(T',\sigma))$. □

Let (G,σ) be an undirected, vertex-colored graph with vertex set L and $|\sigma(L)| = n$. We denote the connected components of (G,σ) by (G_i^n,σ_i^n), $1 \le i \le k$, with vertex sets L_i^n if $\sigma(L_i^n) = \sigma(L)$ and $(G_j^{<n},\sigma_j^{<n})$, $1 \le j \le l$, with vertex sets $L_j^{<n}$ if $\sigma(L_j^{<n}) \subsetneq \sigma(L)$. That is, the upper index distinguishes components with all colors present from those that contain only a proper subset. Suppose that each (G_i^n,σ_i^n) and $(G_j^{<n},\sigma_j^{<n})$ is an RBMG. Then there are trees (T_i^n,σ_i^n) and $(T_j^{<n},\sigma_j^{<n})$ explaining (G_i^n,σ_i^n) and $(G_j^{<n},\sigma_j^{<n})$, respectively. The roots of these trees are u_i and v_j, respectively. We construct a tree (T_G^*,σ) with leaf set L in two steps:

(1) Let (T',σ^n) be the tree obtained by attaching the trees (T_i^n,σ_i^n) with their roots u_i to a common root ρ'.

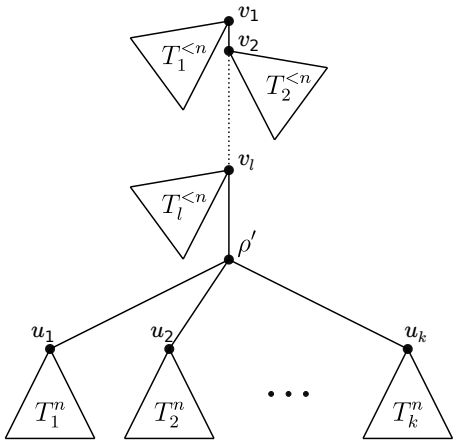


Fig. 22. Shown is a tree (T_G^*, σ) with fork set $\zeta(T_G^*, \sigma) = \{u_1, u_2, \ldots, u_k\}$, that explains the graph $(G, \sigma) = \bigcup_{1 \leq i \leq k} G((T_i^n, \sigma_i^n)) \cup \bigcup_{1 \leq j \leq l} G((T_j^{<n}, \sigma_j^{<n}))$ such that each of the subtrees (T_i^n, σ_i^n) and $(T_j^{<n}, \sigma_j^{<n})$ induces one connected component of (G, σ). The subtree (T_i^n, σ_i^n) with fork u_i is color-complete and explains the n-colored connected component (G_i^n, σ_i^n) of (G, σ). Each connected component $(G_j^{<n}, \sigma_j^{<n})$ that does not contain all colors of S, is explained by a subtree $(T_j^{<n}, \sigma_j^{<n})$. Any n-RBMG (G, σ) with $n \geq 2$ can be explained by a tree of such form (cf. Lemma 5.15). See Fig. 23 for an explicit example of such a tree (T_G^*, σ).

(2) First, construct a path $P = v_1 v_2 \ldots v_{l-1} v_l \rho'$, where ρ' is omitted whenever T' is empty. Now attach the trees $(T_j^{<n}, \sigma_j^{<n})$, $1 \leq j \leq l$, to P by identifying the root of each $T_j^{<n}$ with the vertex v_j in P. Finally, if (T', σ^n) exists, attach it to P by identifying the root of T' with the vertex ρ' in P. The coloring of L is the one given for (G, σ).

This construction is illustrated in Fig. 22 for $n \geq 2$. For $n = 1$, the resulting tree is simply the star tree on L.

Our goal for the remainder of this section is to show that every RBMG is explained by a tree of the form (T_G^*, σ). We start by collecting some useful properties of (T_G^*, σ).

Observation 5.2. *Let (G, σ) be an undirected vertex-colored graph with $|\sigma(V(G))| \geq 2$ whose connected components are RBMGs and let (T_G^*, σ) be the tree described above. Then*

(i) $\zeta(T_G^*, \sigma) = \{u_1, \ldots, u_k\}$,

(ii) Every subtree (T_i^n, σ_i^n), $1 \leq i \leq k$ and $(T^*(v_j), \sigma_{|L(T_G^*(v_j))})$ and $1 \leq j \leq l$, resp., is color-complete.

Proof. Statement (i) is an immediate consequence of Cor. 5.4(ii). For Statement (ii) observe that, by construction, $\sigma(L_i^n) = \sigma(L)$ and thus, (T_i^n, σ_i^n) is a color-complete subtree of (T_G^*, σ), $1 \leq i \leq k$. By Step (2) of the construction of (T_G^*, σ), we have $u_1 \prec_{T_G^*} \rho' \prec_{T_G^*} v_l \prec_{T_G^*} \cdots \prec_{T_G^*} v_1$. Since u_1 is color-complete by assumption, so is each of its ancestors. □

Lemma 5.15. *Let (G, σ) be an undirected vertex-colored graph on n colors whose connected components are RBMGs and there is at least one n-colored*

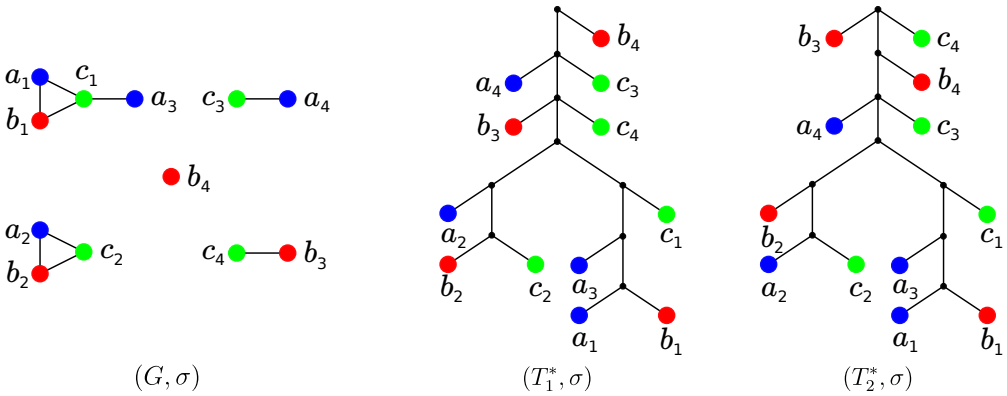


Fig. 23. The trees (T_1^*, σ) and (T_2^*, σ) both explain the 3-RBMG (G, σ) with five connected components and are both of the form (T_G^*, σ).

connected component, and let (T_G^*, σ) *be the tree described above. Then* (T_G^*, σ) *explains* (G, σ).

Proof. For $n = 1$, (T_G^*, σ) is simply the star tree on $V(G)$. Clearly, (G, σ) must be the edge-less graph, which is explained by (T_G^*, σ). Now suppose $n > 1$. Let (G_i^n, σ_i^n) be an n-colored connected component of (G, σ), $i \in \{1, \dots, k\}$ and $k \geq 1$. It has vertex set $L_i^n = L(T_G^*(u_i))$. By construction, $(T_G^*(u_i), \sigma_{|L_i^n}) = (T_i^n, \sigma_i^n)$ explains (G_i^n, σ_i^n) and $(G[L_i^n], \sigma_{|L_i^n}) = (G_i^n, \sigma_i^n)$. Moreover, $(T_G^*(u_i), \sigma_i^n)$ is a color-complete subtree of (T_G^*, σ) that is rooted at u_i. Hence, Lemma 5.12 implies that there are no edges in $G(T_G^*, \sigma)$ between L_i^n and any other vertex in $L \setminus L_i^n$. In other words, (G_i^n, σ_i^n) remains a connected component in $G(T_G^*, \sigma)$, $i \in \{1, \dots, k\}$.

Now suppose that there is a connected component $(G_j^{<n}, \sigma_j^{<n})$, $j \in \{1, \dots, l\}$ and $l \geq 1$, which contains less than n colors. Again, by construction, $(T_G^*(v_j)_{|L_j^{<n}}, \sigma_{|L_j^{<n}}) = (T_j^{<n}, \sigma_j^{<n})$ explains $(G_j^{<n}, \sigma_j^{<n})$ and $(G[L_j^{<n}], \sigma_{|L_j^{<n}}) = (G_j^{<n}, \sigma_j^{<n})$. Furthermore, we have $L_{j'}^{<n} \cap L(T_G^*(v_j)) = \emptyset$ if and only if $j' < j$ by construction of the path $v_1 v_2 \dots v_l$ in T_G^*. By Observation 5.2(ii), v_j is color-complete and Lemma 5.12 implies that there is no edge between $L_j^{<n}$ and any $L_{j'}^{<n}$ whenever $j' < j$. In other words, $(G_j^{<n}, \sigma_j^{<n})$, $j \in \{1, \dots, l\}$ remains a connected component in $G(T_G^*, \sigma)$.

To summarize, *all* connected components of (G, σ) remain connected components in $G(T_G^*, \sigma)$ and are explained by restricting (T_G^*, σ) to the corresponding leaf set, which completes the proof. □

Theorem 5.3. *An undirected leaf-colored graph* (G, σ) *is an* n*-RBMG if and only if each of its connected components is an RBMG and at least one connected component contains all colors.*

Proof. For $n = 1$, the statement trivially follows from the fact that an RBMG must be properly colored and thus, be the edge-less graph for $n = 1$. Now suppose $n > 1$. By Thm. 5.2 every connected component of an RBMG is again an RBMG. Cor. 5.4(i) ensures the existence of a connected component containing all colors. Conversely, if (G, σ) is an undirected graph whose connected components are RBMGs and at least one of them contains all colors, then Lemma 5.15

guarantees that it is explained by a tree of the form (T_G^*, σ) and hence, it is an RBMG. □

The existence of an n-colored connected component is crucial for the statement above. Consider, for instance, an edge-less graph on two vertices, where both vertices have different color. Each of the two connected components is clearly an RBMG, however, one easily checks that their disjoint union is not.

Corollary 5.5. *Every RBMG can be explained by a tree of the form* (T_G^*, σ).

By Thm. 5.3, it suffices to consider each connected component of an RBMG separately. In the following section, therefore, we will consider the characterization of connected RBMGs.

5.5 THREE CLASSES OF CONNECTED 3-RBMGS

In contrast to 2-BMGs, reciprocal best match graphs on two colors convey very little structural information. Their connected components are either single vertices or complete bipartite graphs (cf. Cor. 5.1), which reduce to a K_2 with two distinctly colored vertices under S-thinness. Connected 3-RBMGs, in contrast, can be quite complex. This section aims at giving a complete characterization of 3-RBMGs, which will be later used in Section 5.7 to characterize general n-RBMGs. Moreover, algorithmic results about the recognition and classification of 3-RBMGs as well as the reconstruction of a corresponding tree explaining the RBMG in question will be presented at the end of this section.

5.5.1 *Three Special Classes of Trees*

We start by investigating transformations of trees such that the original tree and the transformed tree both explain the same 3-RBMG. This leads to three types if trees that have a relatively simple structure and we show that any 3-RBMG can be explained by a tree of such type. In the next subsection, this finally results in three disjoint classes of 3-RBMGs that completely cover all possible 3-RBMGs.

The following result shows the possible relocation of a 2-colored subtree (illustrated in Fig. 24) that does not affect the underlying RBMG explained by the original tree:

Lemma 5.16. *Let (G, σ) be an S-thin 3-RBMG that is explained by a tree (T, σ). Moreover, let $u \in V^0(T)$ be a vertex that has two distinct children $v_1, v_2 \in \mathsf{child}(u)$ such that $\sigma(L(T(v_1))) = \sigma(L(T(v_2))) \subsetneq \sigma(L(T))$ and $v_1 \in V^0(T)$, and denote by (T', σ) the tree obtained by replacing the edge uv_2 in (T, σ) by v_1v_2 and possible suppression of u, in case u has exactly two children in (T, σ) or removal of u and its incident edge, in case $u = \rho_{T'}$. Then (T', σ) explains (G, σ).*

Proof. It is easy to see that the resulting tree (T', σ) is phylogenetic. We emphasize that this proof does not depend on whether u has been suppressed

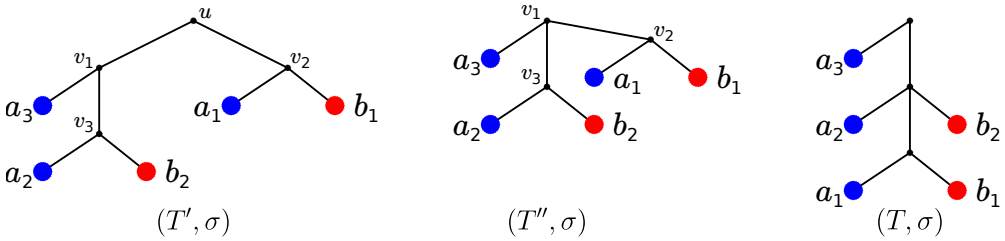


Fig. 24. Assume that the tree (T', σ) is the 2-colored restricted version of some tree that explains a 3-RBMG. According to the transformation of Lemma 5.16, (T'', σ) is obtained from (T', σ) by deletion of the edge uv_2, inserting v_1v_2 and removal of u and its single incident edge. Similarly, (T, σ) is obtained from (T'', σ) by deleting v_1v_2 and inserting v_3v_2. The final tree (T, σ) is a caterpillar. It is easy to verify that all three trees (T', σ), (T'', σ), and (T, σ) explain the same 2-RBMG (G, σ).

or removed. Put $L := L(T)$. Moreover, Lemma 5.7 implies that $L(T(v_1))$ contains leaves of more than one color, hence $|\sigma(L(T(v_1)))| = 2$.

Let $S = \{r, s, t\}$ be the color set of (G, σ) and $\sigma(L(T(v_1))) = \{r, s\}$. Since $L(T(v_1))$ and $L(T(v_2))$ do not contain leaves of color t, we have $\mathrm{lca}_T(y, z) = \mathrm{lca}_{T'}(y, z)$ for every $y \in L[r] \cup L[s]$ and $z \in L[t]$. Hence, $yz \in E(G)$ if and only if $yz \in E(G(T', \sigma))$ for every $y \in L[r] \cup L[s]$ and $z \in L[t]$. It therefore suffices to consider $(T_{rs}, \sigma_{rs}) := (T_{|L[r]\cup L[s]}, \sigma_{|L[r]\cup L[s]})$ and $(T'_{rs}, \sigma_{rs}) := (T'_{|L[r]\cup L[s]}, \sigma_{|L[r]\cup L[s]})$.

First note that, since $T'(v_2) = T(v_2)$, vertex v_2 is color-complete in both (T_{rs}, σ_{rs}) and (T'_{rs}, σ_{rs}). Hence, Lemma 5.12 implies that neither $G(T_{rs}, \sigma_{rs})$ nor $G(T'_{rs}, \sigma_{rs})$ contains edges of the form xy, where $x \in L(T(v_2))$ and $y \notin L(T(v_2))$. Moreover, since $T'(v_2) = T(v_2)$, we have $G(T_{rs}(v_2), \sigma_{|L(T(v_2))}) = G(T'_{rs}(v_2), \sigma_{|L(T(v_2))})$. Since v_1 is also color-complete in (T_{rs}, σ_{rs}) and (T'_{rs}, σ_{rs}), we can similarly conclude that both graphs $G(T_{rs}, \sigma_{rs})$ and $G(T'_{rs}, \sigma_{rs})$ contain no edges xy, where $x \in L(T(v_1))$ and $y \notin L(T(v_1))$. Hence, it suffices to consider edges between leaves in $L(T(v_1))$. If v_1 is a fork in (T_{rs}, σ_{rs}), one can easily see that (T, σ) is obtained from (T', σ) by the same operation used in Lemma 5.14. Hence, Lemma 5.14 implies that $G(T_{rs}, \sigma_{rs}) = (G[L[r] \cup L[s]], \sigma_{rs})$. Suppose that v_1 is not a fork. Note that any $w \in \mathsf{child}_T(v_1)$ with $|\sigma(L(T(w)))| = 1$ must be a leaf as, otherwise, all leaves in $L(T(w))$ would be in a common S-class and (G, σ) would not be S-thin. Therefore, any $w \in \mathsf{child}_T(v_1)$ is either color-complete or a leaf in (T_{rs}, σ_{rs}). Therefore, by Lemma 5.12, there are no edges between $G(T_{rs}(w_1), \sigma_{|L(T(w_1))})$ and $G(T_{rs}(w_2), \sigma_{|L(T(w_2))})$ as soon as one of the children w_1 and w_2 is a non-leaf vertex. In other words, if there are edges between $G(T_{rs}(w_1), \sigma_{|L(T(w_1))})$ and $G(T_{rs}(w_2), \sigma_{|L(T(w_2))})$, then both vertices $w_1, w_2 \in \mathsf{child}_T(v_1)$ are also contained in L. Since, by construction, $\mathsf{child}_T(v_1) \cap L = \mathsf{child}_{T'}(v_1) \cap L$, we have $w_1w_2 \in E(G(T_{rs}, \sigma_{rs}))$ if and only if $w_1w_2 \in E(G(T'_{rs}, \sigma_{rs}))$ for any $w_1, w_2 \in \mathsf{child}_T(v_1)$. Moreover, by construction, we have $(T(w), \sigma_{|L(T(w))}) = (T'(w), \sigma_{|L(T(w))})$ for any inner vertex $w \in \mathsf{child}_T(v_1)$, hence $G(T(w), \sigma_{|L(T(w))}) = G(T'(w), \sigma_{|L(T(w))})$, which concludes the proof.

□

Repeated application of the transformation as in Lemma 5.16 implies the following result, which is illustrated in Fig. 24.

Corollary 5.6. *Let (G,σ) be an S-thin 3-RBMG. Then there exists a tree (T,σ) explaining (G,σ) for which every 2-colored subtree $(T(u),\sigma_{|L(T(u))})$ with $|\sigma(L(T(u)))| = 2$ is a caterpillar and $\sigma(L(T(v_1))) \neq \sigma(L(T(v_2)))$ for any distinct $v_1, v_2 \in \mathsf{child}(u)$.*

Proof. Let (T',σ) explain (G,σ) and let $u \in V^0(T')$ be such that $(T'(u),\sigma_{|L(T'(u))})$ is a 2-colored subtree of (T',σ). Suppose there exists an inner vertex $v \in V^0(T'(u))$ with two distinct children that are again inner vertices, i.e., $w_1, w_2 \in \mathsf{child}_{T'}(v) \cap V^0(T'(u))$. Since (G,σ) is S-thin, we can apply Lemma 5.7 to conclude that $(T'(w_1),\sigma_{|L(T'(w_1))})$ and $(T'(w_2),\sigma_{|L(T'(w_2))})$ are both 2-colored subtrees, thus $\sigma(L(T'(w_1))) = \sigma(L(T'(w_2))) \subsetneq \sigma(L)$. By Lemma 5.16, the tree (T'',σ) that is obtained from (T',σ) by deleting vw_2 and inserting w_1w_2, still explains (G,σ) and satisfies $|\mathsf{child}_{T'}(v) \cap V^0(T''(u))| = |\mathsf{child}_{T'}(v) \cap V^0(T'(u))| - 1$. Repeating this transformation until each inner vertex $v \in V^0(T)$ satisfies $\sigma(L(T(v_1))) \neq \sigma(L(T(v_2)))$ for any $v_1, v_2 \in \mathsf{child}_T(v)$, finally yields a tree (T,σ) for which $|\mathsf{child}(v) \cap V^0(T(u))| \leq 1$, i.e., a caterpillar, that explains (G,σ). In particular, we have $|\sigma(L(T(v_1)))| = 1$ if and only if $v_1 \in L$ (cf. Lemma 5.7), and v cannot have two leaves of the same color as children because (G,σ) is S-thin. □

The restriction to connected S-thin graphs with 3 colors together with the fact that all 2-colored subtrees can be chosen to be caterpillars according to Cor. 5.6 identifies three distinct classes of trees.

Definition 5.8. *Let (T,σ) be a 3-colored tree with color set $S = \{r,s,t\}$. The tree (T,σ) is of*

Type (I) *if there exists $v \in \mathsf{child}(\rho_T)$ such that $|\sigma(L(T(v)))| = 2$ and $\mathsf{child}(\rho_T) \setminus \{v\} \subsetneq L$.*

Type (II) *if there exists $v_1, v_2 \in \mathsf{child}(\rho_T)$ such that $|\sigma(L(T(v_1)))| = |\sigma(L(T(v_2)))| = 2$, $\sigma(L(T(v_1))) \neq \sigma(L(T(v_2)))$ and $\mathsf{child}(\rho_T) \setminus \{v_1, v_2\} \subsetneq L$,*

Type (III) *if there exists $v_1, v_2, v_3 \in \mathsf{child}(\rho_T)$ such that $\sigma(L(T(v_1))) = \{r,s\}$, $\sigma(L(T(v_2))) = \{r,t\}$, $\sigma(L(T(v_3))) = \{s,t\}$, and $\mathsf{child}(\rho_T) \setminus \{v_1, v_2, v_3\} \subsetneq L$.*

An illustration of these three tree types can be found in Fig. 25.

Lemma 5.17. *Let (G,σ) be an S-thin connected 3-RBMG with vertex set L and color set $\sigma(L) = \{r,s,t\}$. Then there is a tree (T,σ) with root ρ_T explaining (G,σ) that satisfies the properties in Cor. 5.6 and is of Type (I), (II), or (III). In particular, all leaves that are incident to the root of (T,σ) must have pairwise distinct colors.*

Proof. Since (G, σ) is an RBMG, there is a tree (T, σ) that explains (G, σ). Denote its root by ρ_T. Note, $|\sigma(L)| = 3$ implies that $|L| \geq 3$.

If $|L| = 3$, then it is easy to see that G must be a complete graph on three vertices. In this case, any tree (T, σ) where T is a triple explains (G, σ) and satisfies Type (I) and Cor. 5.6.

Now suppose $|L| > 3$. Since (G, σ) is connected, Cor. 5.4(ii) implies $\zeta(T, \sigma) = \{\rho_T\}$. Lemma 5.13(i) then implies $L \subseteq \mathcal{L}(\rho_T) \subseteq L$, i.e., $A(\rho_T) = \mathsf{child}(\rho_T)$ and thus, $|\sigma(L(T(v)))| < 3$ for every $v \in \mathsf{child}(\rho_T)$. That is, every proper subtree of (T, σ) contains at most two colors. As a consequence of Cor. 5.6, the tree (T, σ) can be chosen such that there is no pair of distinct vertices $v_1, v_2 \in \mathsf{child}(\rho_T)$ for which $\sigma(L(T(v_1))) = \sigma(L(T(v_2)))$. Moreover, as $|L| > 3$ and $|\sigma(L)| = 3$, it follows directly from Cor. 5.6 that $|\sigma(L(T(v)))| = 1$ for every child $v \in \mathsf{child}(\rho_T)$ is not possible. Thus there is at least one child $v \in \mathsf{child}(\rho_T)$ with $|\sigma(L(T(v)))| \neq 1$ and thus, $|\sigma(L(T(v)))| = 2$.

In summary, there are only six possible subtrees $(T(v), \sigma_{L(T(v))})$ with $v \in \mathsf{child}(\rho_T)$, three containing two colors and three containing only a single color, and each of these six types of subtrees can appear at most once, while there is, in addition, at least one child $v \in \mathsf{child}(\rho_T)$ where $(T(v), \sigma_{L(T(v))})$ contains two colors.

Therefore we end up with the three cases (I), (II), and (III): If there is exactly one vertex $v \in \mathsf{child}(\rho_T)$ such that the subtree $(T(v), \sigma_{L(T(v))})$ contains two colors, any other leaf in $L \setminus L(T(v))$ must be directly attached to ρ_T, thus Condition (I) is satisfied. Similarly, Condition (II) and (III), respectively, correspond to the case where there exist two and three 2-colored subtrees below the root. Since the three types of trees (I), (II), and (III) differ by the number of two-colored subtrees of the root, no tree can belong to more than one type. By the choice of (T, σ), it satisfies Cor. 5.6.

Finally, if the root of a tree is incident to two leaves of the same color, then the graph explained by this tree cannot be S-thin. Thus the last statement must be satisfied. □

The fact that every connected 3-RBMG can be explained by a tree with a very peculiar structure can now be used to infer stringent structural constraints on the 3-RBMGs themselves.

Lemma 5.18. *Let (G, σ) with vertex set L be an S-thin connected 3-RBMG with $\sigma(L) = \{r, s, t\}$ and (T, σ) a tree of Type (I), (II), or (III) explaining (G, σ). Consider $v \in \mathsf{child}(\rho_T)$ such that $\sigma(L(T(v))) = \{r, s\}$. Then:*

(i) If $x \in L(T(v)) \cap L[r]$, then $xy \in E(G)$ for $\sigma(y) = s$ if and only if $\mathsf{par}(x) = \mathsf{par}(y)$ and thus, $y \in L(T(v))$.

If, in addition, there is a vertex $w \in \mathsf{child}(\rho_T) \setminus \{v\}$ with $\sigma(L(T(w))) = \{r, t\}$, i.e., (T, σ) is of either Type (II) or (III), then the following statements hold:

(ii) For any $y \in L(T(v))$, $z \in L(T(w))$, we have $yz \in E(G)$ if and only if $y \in L[s]$ and $z \in L[t]$.

(iii) If (T, σ) is of Type (II), then $yz \in E(G)$ for every $y \in L[s]$ and $z \in L[t]$.

(iv) For any $a \in L(T(v))$, $b \in \mathsf{child}(\rho_T) \cap L$ with $\sigma(b) \neq \sigma(a)$, we have $ab \in E(G)$ if and only if $\sigma(b) \notin \sigma(L(T(v)))$.

Proof. (i) Assume $y \in L[s]$ and $xy \in E(G)$. For contradiction, suppose $y \notin L(T(v))$. Since $L(T(v))$ contains at least one leaf $y' \neq y$ of color s, we have $\mathrm{lca}(x, y') \preceq v \prec \mathrm{lca}(x, y)$, which implies $xy \notin E(G)$; the desired contradiction. Hence, $y \in L(T(v))$. Now assume, again for contradiction, $\mathsf{par}(x) \neq \mathsf{par}(y)$. There are three cases: (a) $\mathsf{par}(x)$ and $\mathsf{par}(y)$ are incomparable in (T, σ), (b) $\mathsf{par}(x) \prec_T \mathsf{par}(y)$, or (c) $\mathsf{par}(y) \prec_T \mathsf{par}(x)$. In Case (a), Lemma 5.7 implies that there is a leaf $y' \in L(T(\mathsf{par}(x)) \cap L[s]$ and therefore, $\mathrm{lca}(x, y') \prec \mathrm{lca}(x, y)$; again a contradiction to $xy \in E(G)$. Similar argumentation can be applied to the Cases (b) and (c). Hence, we conclude that $\mathsf{par}(x) = \mathsf{par}(y)$.
Conversely, assume $\mathsf{par}(x) = \mathsf{par}(y)$ and $y \in L(T(v))$. By construction, we have $\mathsf{par}(x) = \mathrm{lca}(x, y) \preceq \mathrm{lca}(x, y')$ for all $y' \in L[s]$, thus $xy \in E(G)$.
(ii) Let $y \in L(T(v))$, $z \in L(T(w))$, and $yz \in E(G)$. Assume, for contradiction, $\sigma(y) = r$. Since (G, σ) does not contain edges between vertices of the same color, we have $z \in L[t]$. By construction of (T, σ), there must be some $x \in L(T(w))$ of color r. Hence, $\mathrm{lca}(z, x) \preceq w \prec \mathrm{lca}(z, y) = \rho_T$; a contradiction to $yz \in E(G)$. Thus $\sigma(y) = s$. An analogous argument yields $\sigma(z) = t$. Conversely, let $y \in L(T(v))$ and $z \in L(T(w))$ such that $\sigma(y) = s$ and $\sigma(z) = t$. Since neither $t \in \sigma(L(T(v)))$ nor $s \in \sigma(L(T(w)))$ and (T, σ) is of Type (II) or (III), we can immediately conclude that $\mathrm{lca}(y, z) = \rho_T = \mathrm{lca}(y, z') = \mathrm{lca}(y', z)$ for all $y' \in L[s]$ and all $z' \in L[t]$. Thus $yz \in E(G)$.
(iii) Since, $s \notin \sigma(L(T(w)))$, $t \notin \sigma(L(T(v)))$, and (T, σ) is of Type (II), we have $\mathrm{lca}_T(y, z) = \rho_T$ for any pair $y \in L[s]$, $z \in L[t]$. Thus $yz \in E(G)$.
(iv) Let $\sigma(a) = r$ and suppose first $\sigma(b) = s$. Then there is some $y \prec v$ with $\sigma(y) = s$, thus $\mathrm{lca}(a, y) \prec \mathrm{lca}(a, b)$. Therefore a and b cannot be reciprocal best matches, i.e., $ab \notin E(G)$. Now assume $\sigma(b) = t$. Since $t \notin \sigma(L(T(v)))$, we have $\mathrm{lca}(a, z) = \rho_T$ for every $z \in L[t]$. In particular, we have $\mathrm{lca}(b, L[r]) = \rho_T$ and therefore, $ab \in E(G)$.

□

Note that Lemma 5.18(iv) is also satisfied by Type (I) trees. In the following we additionally need a special form of Type (II) trees:

Definition 5.9. *A tree (T, σ) of Type* (II) *with color set $S = \{r, s_1, s_2\}$ and root ρ_T, where $v_1, v_2 \in \mathsf{child}(\rho_T)$ with $\sigma(L(T(v_1))) = \{r, s_1\}$ and $\sigma(L(T(v_2))) = \{r, s_2\}$, is of Type* (II*) *if, for $i \in \{1, 2\}$, it satisfies:*

($\star$) *If there is a vertex $w \in V^0(T(v_i))$ such that $\mathsf{child}(w) \cap L = \{x\}$ for some $x \in L[r]$, then there is a leaf $v \in \mathsf{child}(\rho_T)$ such that $\sigma(v) = s_i$.*

Thus, for leaf-colored trees explaining an S-thin graph, the latter definition, in particular, implies that if there is some vertex $w \in V^0(T(v_i))$ with $\sigma(\mathsf{child}(w) \cap L) = \{r\}$ in a tree (T, σ) of Type (II*), then $L[s_i] \setminus L(T(v_i)) \neq \emptyset$. Furthermore, note that, in a leaf-colored tree explaining an S-thin graph, the property $\sigma(\mathsf{child}(w) \cap L) = \{r\}$ always implies $|\mathsf{child}(w) \cap L| = 1$.

Given an arbitrary tree (T, σ) of Type (II) with colors and subtrees as in Def. 5.9, one can easily construct a corresponding tree (T', σ) of Type (II*) using the following rule for $i \in \{1, 2\}$:

(R) If there is no leaf $v \in \mathsf{child}_T(\rho_T)$ such that $\sigma(v) = s_i$, then re-attach all vertices $x \in L[r]$ with $\mathsf{child}_T(\mathsf{par}_T(x)) \cap L = \{x\}$ to ρ_T and suppress $\mathsf{par}(x)$ in case $\mathsf{par}(x)$ has degree 2 after removal of the edge $\mathsf{par}(x)x$.

By construction, the tree (T', σ) has no vertices $w \in V^0(T(v_i))$ with $\sigma(\mathsf{child}(w) \cap L) = \{r\}$ and thus, (T', σ) trivially satisfies $(\star)$. Hence, (T', σ) is of Type (II*).
We proceed by showing that Rule (R) must be applied to at most one leaf in order to obtain a tree (T', σ) of Type (II*).

Lemma 5.19. *Let (T, σ) be a Type (II) tree that is not of Type (II*) and that explains a connected S-thin 3-RBMG. Let ρ_T be the root of (T, σ) and $S = \{r, s_1, s_2\}$ its color set. Moreover, let $v_1, v_2 \in \mathsf{child}(\rho_T)$ such that $\sigma(L(T(v_i))) = \{r, s_i\}$, $i \in \{1, 2\}$. Then*

(i) no leaf of color r is incident to ρ_T, and

(ii) if Rule (R) is applied to some vertex $x \in L(T(v_i))$, then x is the only leaf in $L[r] \cap L(T(v_i))$ with $\mathsf{child}(\mathsf{par}(x)) \cap L = \{x\}$ and all inner vertices in $L(T(v_j))$, $j \neq i$ satisfy Property $(\star)$ in Def. 5.9.

Proof. First note that, since (T, σ) is of Type (II), it satisfies $\mathsf{child}(\rho_T) \setminus \{v_1, v_2\} \subset L$. Since (T, σ) is not of Type (II*), there must be a leaf $x \in L[r]$ with $w := \mathsf{par}(x) \preceq_T v_i$ and $\sigma(\mathsf{child}(w) \cap L) = \{r\}$ and there is no leaf of color s_i incident to ρ_T, i.e., $L[s_i] \subseteq L(T(v_i))$, for some $i \in \{1, 2\}$. W.l.o.g. we can assume $i = 1$. Now, $L[s_1] \subseteq L(T(v_1))$ and Lemma 5.18(i) implies that $N_{s_1}(x) = \emptyset$ in (G, σ). However, since (G, σ) is connected, there must exist some $z \in L[s_2]$ such that $xz \in E(G)$. Lemma 5.18(i)+(iv) then implies that every $z \in L[s_2]$ with $xz \in E(G)$ must be incident to ρ_T. However, Lemma 5.17 implies that z is the only leaf of color s_2 that is incident to the root. As $N_{s_1}(x) = \emptyset$, it holds that z is the only vertex in L that is adjacent to x in G and thus, $N(x) = \{z\}$ in (G, σ).

In order to show Statement (i), we now assume, for contradiction, that there exists another leaf $x' \neq x$ of color r such that $x' \in \mathsf{child}(\rho_T)$. Then, as a consequence of Lemma 5.17 and since $L[s_1] \subseteq L(T(v_1))$, we have $\mathsf{child}(\rho_T) \cap L = \{x', z\}$. Thus Lemma 5.18(i)+(iv) implies that x' is not adjacent to any vertex in $L(T(v_1)) \cup L(T(v_2))$. Moreover, we have $\mathrm{lca}_T(x', z) = \rho_T = \mathrm{lca}_T(x'', z) = \mathrm{lca}_T(x', z')$ for all $x'' \in L[r]$ and $z' \in L[s_2]$, hence $x'z \in E(G)$. Taking the latter two arguments together with the observation that there is no leaf with color s_1 incident to the root ρ_T, we obtain $N(x') = N(x) = \{z\}$ in (G, σ); a contradiction to the S-thinness of (G, σ).

We proceed with showing Statement (ii). Repeating the latter arguments, one easily checks that $N(x_1) = \{z\}$ for any vertex $x_1 \in L(T(v_1))$ with $\sigma(\mathsf{child}(\mathsf{par}(x_1)) \cap L) = \{r\}$. However, since (G, σ) is S-thin, we cannot have another vertex $x_1 \in L(T(v_1))$ with $N(x_1) = \{z\} = N(x)$. Hence, there is exactly one $x \in L[r] \cap L(T(v_1))$ with $\mathsf{child}(\mathsf{par}(x)) \cap L = \{x\}$ to which Rule (R) can be applied. Moreover, the existence of $z \in \mathsf{child}(\rho_T) \cap L[s_2]$ immediately implies that Property $(\star)$ in Def. 5.9 is satisfied for every $w \in V^0(T(v_2))$ with $\sigma(\mathsf{child}(w) \cap L) = \{r\}$. Thus Statement (ii) is satisfied. □

As a consequence, we can now state the following result:

Lemma 5.20. *If a connected* S*-thin 3-RBMG can be explained by tree of Type* (II)*, then it can be explained by a tree of Type* (II*)*.*

Proof. Assume that (T,σ) is of Type (II) and that $G(T,\sigma)$ is a connected S-thin 3-RBMG. Let $S=\{r,s,t\}$ be the color set of $L:=L(T)$ and $v_1,v_2\in \mathsf{child}(\rho_T)$ with $\sigma(L(T(v_1)))=\{r,s\}$ and $\sigma(L(T(v_2)))=\{r,t\}$. If (T,σ) is already of Type (II*), then the statement is trivially true.

Now suppose that (T,σ) is not of Type (II*). Lemma 5.19 implies that there is exactly one leaf $x\in L[r]$ to which Rule (R) can be applied. Hence, by using Rule (R) and thus re-attaching x to the root, one obtains a tree (T',σ) of Type (II*). In particular, Lemma 5.19 implies that x is the only vertex with color r in (T',σ) incident to the root. W.l.o.g. assume that $x\in L(T(v_1))$. Note, in particular, that the necessity of relocating x implies $L[s]\setminus L(T(v_1))=\emptyset$ (cf. Rule (R)), i.e., $L[s]\subseteq L(T(v_1))$. Thus $\mathsf{child}(\mathsf{par}(x))\cap L=\{x\}$ and Lemma 5.18(i)+(iv) implies that $N_s(x)=\emptyset$ in $G(T,\sigma)$.

Since $L=L(T')$, it suffices to show that $E(G(T',\sigma))=E(G(T,\sigma))$ to prove that (T',σ) explains $G(T,\sigma)$. One easily checks that the only edges that may be different between both sets are those containing the leaf x.

We start by showing $N_s(x)=\emptyset$ in $G(T',\sigma)$. Observe first that, as we have only changed the position of vertex $x\in L[r]$ to obtain (T',σ), $L[s]\subseteq L(T(v_1))$ implies $L[s]\subseteq L(T'(v_1))$. By Lemma 5.7, we have $\sigma(L(T(w)))=\{r,s\}$ for all inner vertices $w\preceq_T v_1$ in T. Thus there must be a vertex $w\in(L(T(v_1)))$ that is incident to two leaves x' and y with $\sigma(x')=r$ and $\sigma(y)=s$. Since $\{x',y\}\subseteq \mathsf{child}(\mathsf{par}(y))$, it follows $x'\neq x$ and thus, x' has not been re-attached. The latter implies that $\sigma(L(T'(v_1)))=\{r,s\}$ and, by construction, $\mathrm{lca}_{T'}(x,y')=\rho_{T'}\succ_{T'} v_1\succ_{T'}\mathrm{lca}_{T'}(x',y')$ for all $x'\in L(T'(v_1))\cap L[r]$ and $y'\in L[s]\cap L(T'(v_1))=L[s]$. Therefore there is no edge between x and any $y'\in L[s]$ in $G(T',\sigma)$. Hence, $N_s(x)=\emptyset$ in $G(T',\sigma)$.

It remains to show that $xz\in E(G(T,\sigma))$ if and only if $xz\in E(G(T',\sigma))$ for this particular re-located vertex x and all $z\in L[t]$. Since $G(T,\sigma)$ is connected and $N_s(x)=\emptyset$, there must exist some vertex z with color t such that $xz\in E(G(T,\sigma))$. Note, Lemma 5.18(ii) implies that there are no edges xz for all $z\in L[t]\cap L(T(v_2))$. That is, x and $z\in L[t]$ form an edge xz in $G(T,\sigma)$ if and only if z is incident to the root ρ_T (cf. Lemma 5.18(ii)+(iv)). In particular, we have by construction of (T',σ) that $\mathrm{lca}_{T'}(x,z)=\rho_{T'}=\mathrm{lca}_{T'}(x',z)=\mathrm{lca}_{T'}(x,z')$ for all $x'\in L[r]$ and all $z'\in L[t]$ and hence, $xz\in E(G(T',\sigma))$.

Now assume that $xz\in E(G(T',\sigma))$ for this particular re-located vertex x and some $z\in L[t]$. Since x has been re-attached, we have $\mathrm{lca}_{T'}(x,z)=\rho_{T'}$. By Lemma 5.19, none of the vertices in $L(T(v_2))$ has been re-attached. Hence, $L(T'(v_2))=L(T(v_2))$ and thus, $\sigma(L(T(v_2)))=\{r,t\}$. Moreover, we have $xz'\notin E(G(T',\sigma))$ for all $z'\in L[t]\cap L(T(v_2))$ since $\mathrm{lca}_{T'}(x,z')=\rho_{T'}\succ_{T'}\mathrm{lca}_{T'}(x',z')$ for all $x'\in L[r]\cap L(T(v_2))$. Thus z must be adjacent to $\rho_{T'}$. By construction, z must be adjacent to ρ_T. As argued above, xz in $G(T,\sigma)$ if and only if z is incident to the root ρ_T. Therefore $xz\in E(G(T,\sigma))$.

In summary, $E(G(T',\sigma))=E(G(T,\sigma))$ and hence, (T',σ) explains $G(T,\sigma)$. □

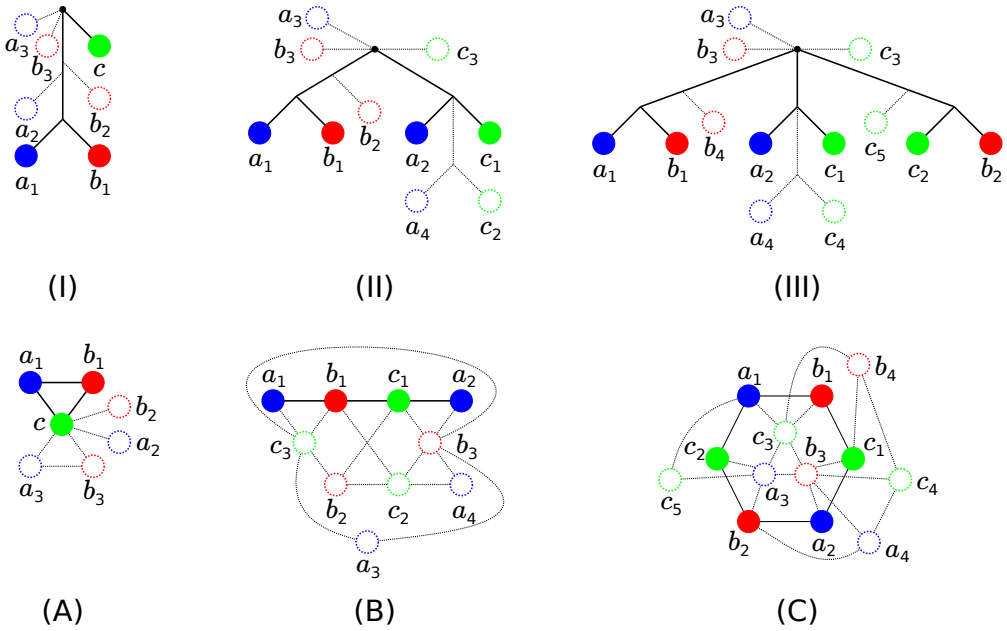


Fig. 25. The three categories of three-colored connected 3-RBMGs are shown on the bottom: (A) Contains a K_3 on three colors but no induced C_n, $n \geq 5$, or P_n, $n \geq 4$, (B) contains an induced P_4, whose endpoints have the same color, but no induced C_n, $n \geq 5$, (C) contains a C_6 of the form (r, s, t, r, s, t). The corresponding tree Types (I), (II), and (III) are shown on top. Solid lines represent edges and vertices that must necessarily be contained in the graph, dashed elements may be missing.

5.5.2 *Three classes of* S*-thin* 3*-RBMGs*

We are now in the position to use these results to show that connected components of 3-RBMGs can be grouped into three disjoint graph classes that correspond to the three tree Types (I), (II), and (III). These three classes are shown in Fig. 25.

Definition 5.10. *An undirected, connected graph (G, σ) on three colors is of*

Type (A) *if (G, σ) contains a K_3 on three colors but no induced P_n for $n \geq 4$, and thus also no induced C_n for $n \geq 5$.*

Type (B) *if (G, σ) contains an induced P_4 on three colors whose endpoints have the same color, but no induced C_n for $n \geq 5$.*

Type (C) *if (G, σ) contains an induced C_6 along which the three colors appear twice in the same permutation, i.e., (r, s, t, r, s, t).*

Theorem 5.4. *Let (G, σ) be an* S*-thin connected 3-RBMG. Then (G, σ) is either of Type* (A)*,* (B)*, or* (C)*. An RBMG of Type* (A)*,* (B)*, and* (C)*, resp., can be explained by a tree of Type* (I)*,* (II)*, and* (III)*, respectively.*

Proof. Let (G, σ) be an S-thin connected 3-RBMG. If $|L| = 3$, then $|\sigma(L)| = 3$ implies that (G, σ) is the complete graph K_3 on three colors, i.e., a graph of Type (A). Every phylogenetic tree on three leaves explains (G, σ), and all of

them except the star are of Type (I). From here on we assume $|L| > 3$. By Lemma 5.17 every connected S-thin 3-RBMG (G, σ) is explained by a tree (T, σ) of either Type (I), (II), or (III).

Claim 1. If (T, σ) is of Type (I), then $(G, \sigma)$is of Type (A).

Proof of Claim 1. Let (T, σ) be a Type (I) tree, i.e., the root ρ_T has one child v such that $\sigma(L(T(v))) = \{r, s\}$ and all other children of ρ_T are leaves. This and $|\sigma(L)| = 3$ implies that there must be a leaf $z \in \mathsf{child}(\rho_T) \cap L[t]$. Since (G, σ) is S-thin, z is the only leaf of color t in (T, σ), thus $xz \in E(G)$ for every $x \in L[r]$ and $yz \in E(G)$ every $y \in L[s]$. In particular, Lemma 5.7 implies that there exists an inner vertex $u \preceq_T v$ such that $\mathsf{child}(u) = \{x^*, y^*\}$ with $x^* \in L[r]$ and $y^* \in L[s]$, thus we have $x^*y^* \in E(G)$. Hence, the induced subgraph $G[x^*y^*z]$ forms a K_3.

It remains to show that (G, σ) contains no induced C_n with $n \geq 5$ or P_n with $n \geq 4$. Since $L[t] = \{z\}$ and $xz, yz \in E(G)$ for any $x \in L[r]$ and $y \in L[s]$, we can conclude that there cannot be any induced P_n, $n \geq 4$, and thus no induced C_n, $n \geq 5$, either, that contains color t. Now assume, for contradiction, that there is an induced P_4 that contains the two colors r, s. By construction, this P_4 must have subsequent coloring (r, s, r, s), thus it contains three distinct vertices x, y_1, y_2 such that $x \in L[r]$, $y_1, y_2 \in L[s]$, and x is adjacent to y_1 and y_2. Lemma 5.18(i) implies that $\mathsf{par}(y_1) = \mathsf{par}(y_2)$. Hence, $N(y_1) = N(y_2)$, which contradicts the S-thinness of (G, σ). Thus there exists no induced P_4, and thus no induced P_n, $n \geq 4$, containing only two colors.

Hence, (G, σ) is of Type (A). ◁

Claim 2. If (T, σ) is of Type (II), then (G, σ) is of Type (B).

Proof of Claim 2. Let (T, σ) be a Type (II) tree, i.e., the root has two distinct children $v_1, v_2 \in \mathsf{child}(\rho_T)$ such that $\sigma(L(T(v_1))) = \{r, s\}$ and $\sigma(L(T(v_2))) = \{r, t\}$, and all other children of the root are leaves.

We start by showing that (G, σ) contains the particular colored induced P_4. Lemma 5.7 implies that there must be a leaf $y_1 \in L(T(v_1)) \cap L[s]$ such that $\mathsf{par}(y_1) = \mathsf{par}(x_1)$ for some $x_1 \in L(T(v_1)) \cap L[r]$ and therefore, $x_1y_1 \in E(G)$. Similarly, there exist two leaves $z_1 \in L(T(v_2)) \cap L[t]$ and $x_2 \in L(T(v_2)) \cap L[r]$ such that $x_2z_1 \in E(G)$. Lemma 5.18(ii) implies $x_1z_1 \notin E(G)$ and $x_2y_1 \notin E(G)$. Clearly, $x_1x_2 \notin E(G)$ since the two vertices have the same color. Moreover, Lemma 5.18(iii) implies $y_1z_1 \in E(G)$. Hence, $\langle x_1y_1z_1x_2 \rangle$ forms an induced P_4 in (G, σ) on three colors whose endpoints have the same color.

We proceed by showing that (G, σ) does not contain an induced C_n with $n \geq 5$. First note that Lemma 5.18(iii) implies $yz \in E(G)$ for any two leaves $y \in L[s]$, $z \in L[t]$. Thus (G, σ) cannot contain an induced C_n for some $n \geq 5$ on colors s and t only. Therefore assume, for contradiction, that there exists an induced C_n for some fixed $n \geq 5$ in $G(T, \sigma)$ that contains a leaf x of color r. Note that this necessarily implies $|N(x)| > 1$ in G. Suppose first that $x \in \mathsf{child}(\rho_T)$. Since $(T(v_1), \sigma_{|L(T(v_1))})$ and $(T(v_2), \sigma_{|L(T(v_2))})$ both contain leaves of color r, any vertex that is adjacent to x in G must be incident to ρ_T in T. Hence, as (G, σ) is S-thin, $|N(x)| > 1$ in G implies that $\mathsf{child}(\rho_T) \cap L = \{x, y, z\}$, where $y \in L[s]$ and $z \in L[t]$, i.e., we have $N(x) = \{y, z\}$. Thus any induced C_n, $n \geq 5$ containing x must also contain both y and z. However, as x, y, and z have the same parent in T, they clearly form a K_3 in G; a contradiction to

x, y, and z being part of an induced C_n. Now suppose $x \in L(T(v_1)) \cap L[r]$. Since $(T(v_2), \sigma_{|L(T(v_2))})$ contains the colors r and t, (G, σ) cannot contain an edge xz with $z \in L(T(v_2))$ (cf. Lemma 5.18(ii)). Hence, $N_t(x) \neq \emptyset$ if and only if there exists a leaf z of color t that is directly attached to the root ρ_T. Since $G(T, \sigma)$ is S-thin, there can be at most one leaf of color t that is attached to ρ_T, thus $|N_t(x)| \leq 1$. This and $|N(x)| > 1$ in G implies that there must be a leaf $y \in L[s]$ such that $y \in N_s(x)$. By Lemma 5.18(i), this is the case if and only if $y \in L(T(v_1)) \cap L[s]$ and $\mathsf{par}(x) = \mathsf{par}(y)$. Since $G(T, \sigma)$ is S-thin, there exists at most one leaf of color s with this property, hence in particular $N(x) = \{y, z\}$. Using Lemma 5.18(iv), we can conclude that $yz \in E(G)$. Thus x, y, and z form a K_3. Therefore these three leaves cannot be contained together in an induced C_n, $n \geq 5$. Since an analogous argumentation holds if $x \in L(T(v_2))$, we conclude that there cannot be an induced C_n, $n \geq 5$, containing a leaf of color r.

In summary, G cannot contain an induced C_n for $n \geq 5$ at all and thus, (G, σ) is of Type (B). ◁

Claim 3. If (T, σ) is of Type (III), then (G, σ) is of Type (C).

Proof of Claim 3. Let (T, σ) be a Type (III) tree, i.e., the root ρ_T has three children $v_1, v_2, v_3 \in \mathsf{child}(\rho_T)$ such that $\sigma(L(T(v_1))) = \{r, s\}$, $\sigma(L(T(v_2))) = \{r, t\}$, and $\sigma(L(T(v_3))) = \{s, t\}$, and all remaining children are leaves. Again, Lemma 5.7 and Lemma 5.18(i) imply that there exist $x_1, y_1 \in L(T(v_1))$ with $x_1y_1 \in E(G)$, $x_2, z_1 \in L(T(v_2))$ with $x_2z_1 \in E(G)$ and $y_2, z_2 \in L(T(v_3))$ with $y_2z_2 \in E(G)$, where $x_i \in L[r]$, $y_i \in L[s]$ and $z_i \in L[t]$. Applying Lemma 5.18(ii), we can in addition conclude that $y_1z_1, x_2y_2, x_1z_2 \in E(G)$, and (G, σ) contains none of the edges x_1z_1, x_1y_2, y_1x_2, y_1z_2, z_1y_2, or x_2z_2. Moreover, (G, σ) does not contain edges between vertices of the same color. Hence, $(G[C], \sigma_{|C})$ with $C = \{x_1, y_1, z_1, x_2, y_2, z_2\}$ forms the desired induced C_6. Therefore (G, σ) is of Type (C). ◁

By definition, the three classes of 3-RBMGs (A), (B), and (C) are disjoint. Lemma 5.17 states that the three classes of trees (I), (II), and (III) are disjoint, hence there is a one-to-one correspondence between the tree Types (I), (II), and (III) and the graph classes (A), (B), and (C). □

An undirected, colored graph (G, σ) contains an induced K_3, P_4, or C_6, respectively, if and only if $(G/\mathsf{S}, \sigma/\mathsf{S})$ contains an induced K_3, P_4, or C_6, resp., on the same colors (cf. Lemma 5.4). An immediate consequence of this fact is

Theorem 5.5. *A connected (not necessarily S-thin) 3-RBMG (G, σ) is either of Type* (A)*,* (B)*, or* (C)*.*

In the following subsections we characterize the three classes of 3-RBMGs, starting with Type (A).

5.5.3 *Characterization of Type* (A) *3-RBMGs*

As an immediate consequence of Thm. 5.4 and the well-known property that cographs are exactly those undirected graphs without induced P_4s, we immediately obtain an characterization of 3-RBMGs that are of Type (A):

Observation 5.3. *Let (G, σ) be a connected, S-thin 3-RBMG. Then it is of Type* (A) *if and only if it is a cograph.*

However, this characterization requires the prior knowledge if a given graph (G, σ) is an 3-RBMG or not. Another, more general, characterization can be obtained by means of a so-called hub-vertex.

Definition 5.11. *Let $G = (V, E)$ be an undirected graph. A vertex $x \in V(G)$ such that $N(x) = V \setminus \{x\}$ is a* hub-vertex.

Lemma 5.21. *A properly vertex-colored, connected, S-thin graph (G, σ) on three colors with vertex set L is a 3-RBMG of Type* (A) *if and only if $G \notin \mathcal{P}_3$ and it satisfies the following conditions:*

(A1) *G contains a hub-vertex x, i.e., $N(x) = V(G) \setminus \{x\}$*

(A2) *$|N(y)| < 3$ for every $y \in V(G) \setminus \{x\}$.*

Proof. By definition, a 3-RBMG is properly colored and has $|L| \geq 3$ vertices. If $|L| = 3$ there are only two connected graphs: K_3 and P_3. Both satisfy (A1) and (A2) since the three vertices have distinct colors. However, only K_3 is a 3-RBMG: it is explained by any tree on three leaves with pairwise distinct colors. From here on we assume $|L| \geq 4$.

We start with the "only-if-direction" and show that every 3-RBMG of Type (A) satisfies (A1) and (A2). We set $S = \{r, s, t\}$ and assume that (G, σ) is a 3-RBMG of Type (A). Thm. 5.4 implies that there exists a tree (T, σ) with root ρ_T explaining (G, σ) that is of Type (I), i.e., there is a vertex $v \in \mathsf{child}(\rho_T)$ such that $\sigma(L(T(v))) = \{s, t\}$ and $\mathsf{child}(\rho_T) \setminus \{v\} \subset L$. Thus every leaf x with color $\sigma(x) = r$ is a child of ρ_T. Since (G, σ) is S-thin, this implies $|L[r]| = 1$ and therefore, $xy \in E(G)$ for every $y \neq x$. Hence, (A1) is satisfied. In order to show (A2), consider $y \in V(G) \setminus \{x\}$, where x is again the unique vertex with color r. Since (G, σ) is properly colored, we have $\sigma(y) \neq r$. W.l.o.g., let $\sigma(y) = s$. Assume, for contradiction, that $|N(y)| \geq 3$. Then there are at least two distinct vertices $z, z' \in N_t(y)$. Assume first that $y \in L(T(v))$. Thus there exists $z^* \in L[t]$ with $\mathrm{lca}(y, z^*) \preceq v \prec \rho_T$. Hence, we must have $z, z' \in L(T(v))$. However, Lemma 5.18(i) implies that z and z' must be siblings and therefore $N(z) = N(z')$; a contradiction since (G, σ) was assumed to be S-thin. Now assume that $y \in \mathsf{child}(\rho_T)$. Then Lemma 5.18(iv) and $z, z' \in N_t(y)$ imply that z and z' both have to be adjacent to ρ_T; again this contradicts the assumption that (G, σ) is S-thin. Thus (A2) is satisfied.

We proceed with showing the "if-direction". Suppose (G, σ) is a properly vertex-colored, connected, S-thin graph satisfying (A1) and (A2). In order to show that (G, σ) is a Type (A) RBMG it suffices, by Thm. 5.4, to construct a Type (I) tree that explains (G, σ). Let x be a vertex that is adjacent to all others, which exists by (A1). Assume w.l.o.g. that $\sigma(x) = r$. Since (G, σ) is S-thin and it does not contain edges between vertices of the same color, x must be the only vertex of color r. We define $L_2 := \{y \mid y \neq x, |N(y)| = 2\}$. Since $|L| > 3$ and thus $|N(x)| \geq 3$, we have $x \in L \setminus L_2$. Note that each vertex is adjacent to x and thus, $N(y) = \{x, z\}$ for all $y \in L_2$ and some vertex $z \in L[t]$, $t \neq \sigma(y)$. Property (A2) implies that there are $|L \setminus L_2| - 1$ vertices with degree

1, all incident to x. Since (G, σ) is S-thin and $|S| = 3$, there are at most two vertices with degree 1, at most one of each color different from $\sigma(x)$, and thus $|L \setminus L_2| \leq 3$.

We first construct a caterpillar $(T_2, \sigma_{|L_2})$ with leaf set L_2 and root ρ_{T_2} such that $\mathsf{par}(y) = \mathsf{par}(z)$ for any $y, z \in L_2$ with $\sigma(y) \neq \sigma(z)$ if only if $yz \in E(G)$. As $N(y) = \{x, z\}$ for all $y \in L_2$ and some vertex $z \in L[t]$, $t \neq \sigma(y)$, we can conclude that each connected component of $(G[L_2], \sigma_{|L_2})$ is a single edge yz. Thus it is easy to see that $(T_2, \sigma_{|L_2})$ explains $(G[L_2], \sigma_{|L_2})$.

For the construction of (T, σ), we then distinguish two cases: (i) If $L \setminus L_2 = \{x, w_1\}$, then (T, σ) is obtained by attaching the vertices x and w_1 as well as ρ_{T_2} as children of the root ρ_T. (ii) If $L \setminus L_2 = \{x, w_1, w_2\}$ for distinct vertices x, w_1, and w_2 in (G, σ), we first build an auxiliary tree $(T', \sigma_{|L_2 \cup \{w_2\}})$ with root $\rho_{T'}$ by attaching ρ_{T_2} and vertex w_2 to $\rho_{T'}$. The tree (T, σ) is then constructed from $(T', \sigma_{|L_2 \cup \{w_2\}})$ by attaching x, the other vertex w_1 and $\rho_{T'}$ as children of ρ_T. It remains to show that (T, σ) explains (G, σ). By construction, (T, σ) is a tree of Type (I) where the vertices ρ_{T_2} and $\rho_{T'}$ play the role of v in Def. 5.9 in Case (i) and (ii), respectively. In the following let $v = \rho_{T_2}$ or $v = \rho_{T'}$ depending on whether we have Case (i) and (ii).

Thm. 5.4 implies that $G(T, \sigma)$ is 3-RBMG of Type (A). It is easy to see that $G(T, \sigma)[L_2] = (G[L_2], \sigma_{|L_2})$. Any remaining edges in $G(T, \sigma)$ are thus adjacent to vertices in $L \setminus L_2$. We first consider edges that may be incident to vertex x in $G(T, \sigma)$. Since $\mathrm{lca}_T(z, x) = \rho_T$ and $r \notin \sigma(L(T(v)))$, we have $xz \in E(G(T, \sigma))$ for all $z \in L \setminus \{x\}$. Hence, (A1) is satisfied by x in $G(T, \sigma)$.

Now consider edges that may be incident to vertex w_1 in $G(T, \sigma)$. First note that in both Cases (i) and (ii) the vertex w_1 is adjacent to the root ρ_T in (T, σ). Since (T, σ) is a tree of Type (I), we can apply Lemma 5.18(i) to conclude that there are no edges in $G(T, \sigma)$ between w_1 and any vertex in $L(T(v))$. This and the arguments above show that $N(w_1) = \{x\}$. In other words, $w_1 z \in E(G(T, \sigma))$ if and only if $w_1 z \in E(G)$ for all $z \in L$. This in particular shows $(G, \sigma) = G(T, \sigma)$, which conforms to Case (i).

Finally, assume Case (ii) and consider edges that are incident to vertex w_2 in (G, σ). By construction, $s, t \in \sigma(L_2)$. Since $\mathrm{lca}(w_2, z) = v \succ_T \rho_{T_2} \succeq \mathrm{lca}(w', z)$ for all $w', z \in L_2$ with $\sigma(w_2) = \sigma(w) \neq \sigma(z)$, we can conclude that w_2 is not adjacent to any other vertex in L_2. This and the arguments above show that $N(w_2) = \{x\}$. Therefore $w_2 z \in E(G(T, \sigma))$ if and only if $w_1 z \in E(G)$ for all $z \in L$.

In summary, $G(T, \sigma) = (G, \sigma)$. Therefore (G, σ) is of Type (A). □

The next result is an immediate consequence of Observation 5.3 and Lemma 5.21. We still present a short alternative proof which sheds some more light on the explicit structure of S graphs satisfying (A1) and (A2).

Lemma 5.22. *Let (G, σ) be an S-thin graph satisfying* (A1) *and* (A2)*. Then G is a cograph.*

Proof. Since (G, σ) contains a hub-vertex by (A1), it can be written as join $G' \triangledown K_1$, where K_1 corresponds to the hub-vertex. As a consequence of (A2), G' is a 2-colored graph with vertex degree at most 1. The number of isolated vertices in G' cannot exceed 2, one of each color, since otherwise two vertices

that are isolated in G' would have the same color and thus, share the hub-vertex as their only neighbor in G, contradicting S-thinness of (G,σ). Hence, G' is the disjoint union of an arbitrary number of K_2 and at most two copies of K_1: $G = ((\bigcup^{n_1} K_1) \cup (\bigcup^{n_2} K_2)) \triangledown K_1$ with $0 \leq n_1 \leq 2$ and $n_2 \geq 0$. Thus G is a cograph [38]. □

For later reference, we close this section with a simple property of hub-vertices.

Corollary 5.7. *Let x be a hub-vertex of some connected S-thin 3-RBMG (G,σ) of Type* (A) *with vertex set L and $|L| > 3$. Then x is the only vertex of its color in (G,σ), i.e., $L[\sigma(x)] = \{x\}$. Moreover, for any (T,σ) explaining (G,σ), x must be incident to the root of T.*

Proof. Since (G,σ) does not contain edges between vertices of the same color, the first statement immediately follows from Property (A1) and S-thinness of (G,σ).

For the second statement, let (T,σ) be an arbitrary tree with root ρ_T that explains (G,σ). Let $v \in \mathsf{child}(\rho_T)$ with $x \preceq_T v$. Assume, for contradiction, $v \neq x$. Thus Lemma 5.7 implies that there exists a leaf $y \in L$ with $\sigma(y) \neq \sigma(x)$ such that $y \preceq_T v$. Then, since x is connected to any vertex in $L \setminus \{x\}$, all vertices of color $\sigma(y)$ must be contained in the subtree $T(v)$; otherwise $\mathrm{lca}_T(x,y) \prec_T \mathrm{lca}_T(x,y') = \rho_T$ for some vertex $y' \in L[\sigma(y)]$, $y' \neq y$, which yields a contradiction to $xy' \in E(G)$. As (T,σ) is phylogenetic, the root ρ_T has at least two different children, i.e., there is some $w \in \mathsf{child}(\rho_T)$, $w \neq v$. Let $r \neq \sigma(x), \sigma(y)$ be the third color in (G,σ). We already argued $\sigma(x), \sigma(y) \notin \sigma(L(T(w)))$, thus $\sigma(L(T(w))) = \{r\}$. In particular, since (G,σ) is S-thin, Lemma 5.7 implies that w must be a leaf. Since (G,σ) is connected, we can apply the same arguments as for $L[\sigma(y)]$ to conclude that $r \notin \sigma(L(T(v)))$, thus $|L[r]| = 1$. Since x is the only leaf of its color in (T,σ) and $\sigma(L(T(v))) = \{\sigma(x), \sigma(y)\}$, we can again apply Lemma 5.7 to conclude that $|L[\sigma(y)]| = 1$. In summary, we have therefore shown $|L| = 3$; a contradiction. Hence, x must be incident to ρ_T. □

5.5.4 *Characterization of Type* (B) *3-RBMGs*

This subsection disentangles the structure of (B) 3-RBMGs, which turns out to be more complex than that of Type (A) 3-RBMGs. We start by introducing the notion of *B-like* colored graphs:

Definition 5.12. *Let (G,σ) be an undirected, connected, properly colored, S-thin graph with vertex set L and color set $\sigma(L) = \{r,s,t\}$, and assume that (G,σ) contains the induced path $P := \langle \hat{x}_1\hat{y}\hat{z}\hat{x}_2\rangle$ with $\sigma(\hat{x}_1) = \sigma(\hat{x}_2) = r$, $\sigma(\hat{y}) = s$, and $\sigma(\hat{z}) = t$. Then (G,σ) is* B-like w.r.t. *P if (i) $N_r(\hat{y}) \cap N_r(\hat{z}) = \emptyset$, and (ii) G does not contain an induced cycle C_n, $n \geq 5$.*

For a 3-colored, S-thin graph (G,σ) that is B-like w.r.t. the induced path $P := \langle \hat{x}_1\hat{y}\hat{z}\hat{x}_2\rangle$, we define the following subsets of vertices:

$$\begin{aligned}
L^P_{t,s} :=& \{y \mid \langle xy\hat{z}\rangle \in \mathcal{P}_3 \text{ for any } x \in N_r(y)\}\\
L^P_{t,r} :=& \{x \mid N_r(y) = \{x\} \text{ and } \langle xy\hat{z}\rangle \in \mathcal{P}_3\}\cup\\
&\{x \mid x \in L[r],\, N_s(x) = \emptyset,\, L[s] \setminus L^P_{t,s} \neq \emptyset\}\\
L^P_{s,t} :=& \{z \mid \langle xz\hat{y}\rangle \in \mathcal{P}_3 \text{ for any } x \in N_r(z)\}\\
L^P_{s,r} :=& \{x \mid N_r(z) = \{x\} \text{ and } xz\hat{y} \in \mathcal{P}_3\}\cup\\
&\{x \mid x \in L[r], N_t(x) = \emptyset, L[t] \setminus L^P_{s,t} \neq \emptyset\}
\end{aligned}$$

The first subscripts t and s refer to the color of the vertices $\hat{z}$ and $\hat{y}$, respectively, that "anchor" the P_3s within the defining path P. The second index identifies the color of the vertices in the respective set since, by definition, we have $L^P_{t,s} \subseteq L[s]$, $L^P_{t,r} \subseteq L[r]$, $L^P_{s,t} \subseteq L[t]$ and $L^P_{s,r} \subseteq L[r]$. Furthermore, we set

$$\begin{aligned}
L^P_t :=& L^P_{t,s} \cup L^P_{t,r}\\
L^P_s :=& L^P_{s,t} \cup L^P_{s,r}\\
L^P_* :=& L \setminus (L^P_t \cup L^P_s).
\end{aligned}$$

By definition, $L^P_{s,r} = L^P_s \cap L[r]$, $L^P_{t,r} = L^P_t \cap L[r]$, $L^P_{s,t} = L^P_s \cap L[t]$, and $L^P_{t,s} = L^P_t \cap L[s]$. For simplicity we will often write $L^P_*[i] := L^P_* \cap L[i]$ for $i \in \{s,t\}$. These vertex sets arise naturally from trees of Type (II*):

Lemma 5.23. *Let (G,σ) be a connected, S-thin 3-RBMG of Type* (B) *with vertex set L and color set $S = \{r,s,t\}$. Then the colors can be permuted such that there are $\hat{x}_1, \hat{x}_2 \in L[r]$, $\hat{y} \in L[s]$, $\hat{z} \in L[t]$ such that (G,σ) is B-like w.r.t. $P = \langle \hat{x}_1\hat{y}\hat{z}\hat{x}_2\rangle$. Moreover, there exists a tree (T,σ) of Type* (II*) *explaining (G,σ) such that*

(i) $L^P_t = L(T(v_1))$ *and* $L^P_s = L(T(v_2))$ *for* $v_1, v_2 \in \mathsf{child}(\rho_T) \setminus L$, *and*

(ii) $L^P_* = \mathsf{child}(\rho_T) \cap L$.

Proof. Let (G,σ) be a connected, S-thin 3-RBMG of Type (B). Then, by Lemmas 5.17 and 5.20, there is a tree (T,σ) with root ρ_T explaining (G,σ) that is of Type (II*). In particular, the colors can be chosen such that there are $v_1, v_2 \in \mathsf{child}(\rho_T)$ with $\sigma(L(T(v_1))) = \{r,s\}$, $\sigma(L(T(v_2))) = \{r,t\}$, and $\mathsf{child}(\rho_T) \setminus \{v_1, v_2\} \subset L$. Applying the same argumentation as in the proof of Thm. 5.4 (Claim 2), we conclude that there are leaves $\hat{x}_1, \hat{y} \prec_T v_1$ and $\hat{x}_2, \hat{z} \prec_T v_2$, where $\hat{x}_1, \hat{x}_2 \in L[r]$, $\hat{y} \in L[s]$, $\hat{z} \in L[t]$, such that $\langle \hat{x}_1\hat{y}\hat{z}\hat{x}_2\rangle$ is an induced P_4 in G. By S-thinness of (G,σ) and Lemma 5.18(i), we have $N_r(\hat{y}) = \{\hat{x}_1\}$ and $N_r(\hat{z}) = \{\hat{x}_2\}$ and thus, $N_r(\hat{y}) \cap N_r(\hat{z}) = \emptyset$. Since 3-RBMGs of Type (B) do not contain induced cycles on more than five vertices, (G,σ) is B-like w.r.t. $\langle \hat{x}_1\hat{y}\hat{z}\hat{x}_2\rangle$. It remains to show Properties (i) and (ii).

In the following, we put for simplicity $L_1 := L^P_t$. To establish Property (i) we treat vertices of colors s and r separately. Consider $y \in L[s]$. We first show that $y \in L(T(v_1))$ implies $y \in L_1$. Clearly, since $L(T(v_1))$ contains leaves of color

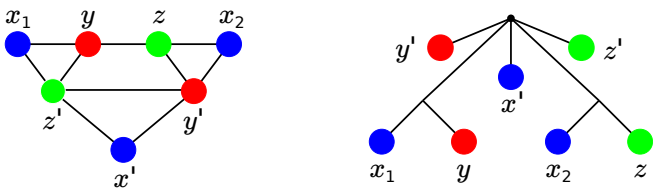
r, any $x \in N_r(y)$ must satisfy $x \prec_T v_1$. Lemma 5.18(i) implies that $x \in N_r(y)$ if and only if $\mathsf{par}(x) = \mathsf{par}(y)$. Therefore $|N_r(y)| \leq 1$ because (G, σ) is S-thin. If $N_r(y) = \emptyset$, then $y \in L_1$ by definition. Otherwise, $N_r(y) = \{x\}$ with $x \prec_T v_1$. By construction of (T, σ), there is a leaf $x' \prec_T v_2$ of color r; this implies $\mathrm{lca}(\hat{z}, x) \succ_T \mathrm{lca}(\hat{z}, x')$ and hence, $x\hat{z} \notin E(G)$. Since $y\hat{z} \in E(G)$ by Lemma 5.18(iii), we have $\langle xy\hat{z}\rangle \in \mathcal{P}_3$ and thus, $y \in L_1$. Hence, $L(T(v_1)) \cap L[s] \subseteq L_1 \cap L[s]$ as claimed. We now show that $y \in L_1$ implies $y \in L(T(v_1))$. To this end, consider $y \in L_1 \cap L[s]$, i.e., either $N_r(y) = \emptyset$ or $\langle xy\hat{z}\rangle \in \mathcal{P}_3$ for every $x \in N_r(y)$. Assume, for contradiction, that $y \notin L(T(v_1))$. Then y must be incident to the root ρ_T. Since $L(T(v_2))$ contains no leaf of color s, Lemma 5.18(iv) implies $y\hat{x}_2, y\hat{z} \in E(G)$. Since $\hat{x}_2\hat{z} \in E(G)$, the vertices $\hat{x}_2, y, \hat{z}$ induce a K_3, thus $\langle \hat{x}_2 y\hat{z}\rangle \notin \mathcal{P}_3$ and therefore, $y \notin L_1$; a contradiction. Hence, we can conclude $L_1 \cap L[s] \subseteq L(T(v_1)) \cap L[s]$. In summary, we therefore have $L(T(v_1)) \cap L[s] = L_1 \cap L[s]$.

Consider $x \in L[r]$. We first show that $x \in L(T(v_1))$ implies $x \in L_1$. If there exists a leaf $y \in L[s]$ incident to $\mathsf{par}(x)$, then $N_r(y) = \{x\}$ by Lemma 5.18(i) and $\langle xy\hat{z}\rangle \in \mathcal{P}_3$ by Lemma 5.18(ii)+(iii), implying $x \in L_1$. Otherwise, S-thinness of G implies that $\mathsf{child}(\mathsf{par}(x)) \cap L = \{x\}$. In this case, $N_s(x) = \emptyset$ by Lemma 5.18(i). Moreover, since (T, σ) is of Type (II*) and $\mathsf{child}(\mathsf{par}(x)) \cap L = \{x\}$, we can apply Condition $(\star)$ in Def. 5.9 to conclude $L[s] \setminus L^P_{t,s} = L[s] \setminus L(T(v_1)) \neq \emptyset$, where equality holds because $L(T(v_1)) \cap L[s] = L_1 \cap L[s] = L^P_{t,s}$. In summary, we have thus shown that $x \in L_1$. Hence, $L(T(v_1)) \cap L[r] \subseteq L_1 \cap L[r]$ as claimed. Conversely, we show that $x \in L_1$ implies $x \in L(T(v_1))$. Assume that $x \in L_1 \cap L[r]$. Then, by definition of L_1, we have $x \in L^P_{t,r}$. Thus either (a) there is a leaf $y \in L[s]$ such that $N_r(y) = \{x\}$ and $\langle xy\hat{z}\rangle \in \mathcal{P}_3$, or (b) $N_s(x) = \emptyset$ and $L[s] \setminus L^P_{t,s} = L[s] \setminus L(T(v_1)) \neq \emptyset$. In Case (a), assume first, for contradiction, that y is adjacent to the root ρ_T. Lemma 5.18(iv) implies that $x'y \in E(G)$ for any $x' \in L(T(v_2)) \cap L[r]$. Since $L(T(v_2)) \cap L[r] \neq \emptyset$ and $|N_r(y)| = 1$, we have $L(T(v_2)) \cap L[r] = \{x\}$ and thus, as $\hat{x}_2 \in L(T(v_2))$, it follows $x = \hat{x}_2$. However, since $x = \hat{x}_2$ and $\hat{z}$ are adjacent in (G, σ), $xy\hat{z}$ cannot form an induced P_3. We therefore conclude that y cannot be adjacent to the root ρ_T. Since $s \notin \sigma(L(T(v_2)))$, it must thus hold $y \in L(T(v_1))$. Lemma 5.18(ii)+(iv) then implies $x \in L(T(v_1))$. In Case (b), $L[s] \setminus L^P_{t,s} = L[s] \setminus L(T(v_1)) \neq \emptyset$ implies that there exists a leaf $y^* \in L[s] \setminus L(T(v_1))$. Since $s \notin \sigma(L(T(v_2)))$, this vertex y^* must be incident to the root ρ_T. On the other hand, we have $xy^* \notin E(G)$ because $N_s(x) = \emptyset$, hence x cannot be incident to ρ_T. Applying Lemma 5.18(iv) thus implies $x \in L(T(v_1))$. Therefore, $L_1 \cap L[r] = L(T(v_1)) \cap L[r]$.

In summary we have shown $L_1 = L(T(v_1))$. By symmetry of the definitions, analogous arguments imply $L^P_s = L(T(v_2))$, completing the proof of statement (i). Property (ii) now immediately follows from $\mathsf{child}(\rho_T) \cap L = L \setminus (L(T(v_1)) \cup L(T(v_2))) = L \setminus (L^P_t \cup L^P_s)$.

□

The following remark will be useful for the design of algorithms to recognize Type (B) RBMGs. It implies, in particular, that testing whether (G, σ) is B-like w.r.t. some induced P_4 strongly depends on the reference P_4, i.e., it is necessary to identify all P_4s in (G, σ).



(G, σ) (T, σ)

Fig. 26. The graph (G, σ) is a 3-RBMG since it is explained by (T, σ). Moreover, (G, σ) does not contain an induced C_n, $n \geq 5$, but induced P_4s, thus it is of Type (B). It is easy to see that (G, σ) is B-like w.r.t. $\langle x_1 y z x_2 \rangle$. However, (G, σ) is not B-like w.r.t. $\langle x_1 z' y' x_2 \rangle$ since $x' \in N_r(y') \cap N_r(z')$.

Observation 5.4. *A connected,* S*-thin 3-RBMG (G, σ) of Type* (B) *may contain distinct induced P_4s P and P', both of the form (r, s, t, r) for distinct colors r, s, t such that (G, σ) is B-like w.r.t. P but not B-like w.r.t. P'. An example is given in Fig. 26.*

Using the previous result, we obtain the following characterization for 3-colored RBMGs of Type (B).

Lemma 5.24. *Let (G, σ) be an undirected, connected,* S*-thin, and properly 3-colored graph with color set $S = \{r, s, t\}$ and let $x \in L[r]$, $y \in L[s]$, and $z \in L[t]$. Then (G, σ) is a 3-RBMG of Type* (B) *if and only if the following conditions are satisfied, after possible permutation of the colors:*

(B1) *(G, σ) is B-like w.r.t. $P = \langle \hat{x}_1 \hat{y} \hat{z} \hat{x}_2 \rangle$ for some $\hat{x}_1, \hat{x}_2 \in L[r]$, $\hat{y} \in L[s]$, $\hat{z} \in L[t]$,*

(B2.a) *If $x \in L_*^P$, then $N(x) = L_*^P \setminus \{x\}$,*

(B2.b) *If $x \in L_t^P$, then $N_s(x) \subset L_t^P$ and $|N_s(x)| \leq 1$, and $N_t(x) = L_*^P[t]$,*

(B2.c) *If $x \in L_s^P$, then $N_t(x) \subset L_s^P$ and $|N_t(x)| \leq 1$, and $N_s(x) = L_*^P[s]$*

(B3.a) *If $y \in L_*^P$, then $N(y) = L_s^P \cup (L_*^P \setminus \{y\})$,*

(B3.b) *If $y \in L_t^P$, then $N_r(y) \subset L_t^P$ and $|N_r(y)| \leq 1$, and $N_t(y) = L[t]$,*

(B4.a) *If $z \in L_*^P$, then $N(z) = L_t^P \cup (L_*^P \setminus \{z\})$,*

(B4.b) *If $z \in L_s^P$, then $N_r(z) \subset L_s^P$ and $|N_r(z)| \leq 1$, and $N_s(z) = L[s]$.*

In particular, L_t^P, L_s^P, and L_*^P are pairwise disjoint and $\hat{x}_1, \hat{y} \in L_t^P$, $\hat{x}_2, \hat{z} \in L_s^P$.

Proof. Suppose first that (G, σ) satisfies Conditions (B1) - (B4.b). By Condition (B1), (G, σ) is B-like, thus in particular it contains no induced C_n with $n \geq 5$. Therefore, if (G, σ) is an RBMG, then it must be of Type (B).

In order to prove that (G, σ) is indeed an RBMG, we construct a tree (T, σ) based on the sets L_t^P, L_s^P, and L_*^P and show that it explains (G, σ). To this end, we show first that the sets L_t^P, L_s^P, and L_*^P are pairwise disjoint. By definition, L_*^P is disjoint from L_t^P and L_s^P. Moreover, by definition, $\sigma(L_t^P) \cap \sigma(L_s^P) = \{r\}$.

Thus it suffices to show that any vertex $x \in L[r] \setminus L_*^P$ is is contained in exactly one of the sets L_t^P or L_s^P. Assume, for contradiction, that there exists a leaf x that is contained in both L_t^P and L_s^P. Hence, $x \in L_{t,r}^P \cap L_{s,r}^P$. Then, by definition of $L_{t,r}^P$, either (a) $N_s(x) = \emptyset$ and $L[s] \setminus L_{t,s}^P \neq \emptyset$, or (b) $N_r(y) = \{x\}$ and $\langle xy\hat{z}\rangle \in \mathcal{P}_3$ for some $y \in L[s]$. Suppose first Case (a). Since $N_s(x) = \emptyset$ and (G, σ) is connected, there must be a vertex $z \in L[t]$ such that $xz \in E(G)$. As $x \in L_s^P$, Condition (B2.c) implies $N_t(x) = \{z\}$. Furthermore, by Condition (B2.c), $N_t(x) \subset L_s^P$ and thus, $z \in L_s^P$. On the other hand, $x \in L_t^P$ and (B2.b) imply $z \in L_*^P$; a contradiction since L_s^P and L_*^P are disjoint. Analogously, in Case (b), Condition (B2.b) implies $y \in L_t^P$, whereas $y \in L_*^P$ by Condition (B2.c), which again yields a contradiction. We therefore conclude that L_t^P, L_s^P, and L_*^P are disjoint.

Moreover, for the construction of (T, σ), we show that $G[L_i^P]$ is the disjoint union of an arbitrary number of K_1s and K_2s with $i \in \{s, t\}$. By definition of L_t^P, we have $\sigma(L_t^P) \subseteq \{r, s\}$. In addition, we have $N_s(x) \subset L_t^P$ and $|N_s(x)| \leq 1$ for any $x \in L_{t,r}^P$ by (B2.b) as well as $N_r(y) \subset L_t^P$ and $|N_r(y)| \leq 1$ for any $y \in L_{t,s}^P$ by (B3.b). Therefore any vertex of L_t^P has at most one neighbor in L_t^P. Similar arguments and application of Properties (B2.c), resp., (B4.b) show that any vertex of L_s^P has at most one neighbor in L_s^P. Thus $G[L_i^P]$ is the disjoint union of an arbitrary number of K_1s and K_2s with $i \in \{s, t\}$.

We are now in the position to construct a tree (T, σ) based on the sets L_t^P, L_s^P, and L_*^P and to show that it explains (G, σ). First, for $i \in \{s, t\}$, we construct a caterpillar (T_i, σ_i), with root ρ_{T_i}, on the leaf set L_i^P such that $\mathsf{par}(a) = \mathsf{par}(b)$ for any $a, b \in L_i^P$ with $\sigma(a) \neq \sigma(b)$ if and only if $ab \in E(G)$. Since $G[L_i^P]$ is the disjoint union of an arbitrary number of K_1s and K_2s, the tree T_i is well-defined. It is, however, not unique as the order of inner vertices in T_i is arbitrary. Then (T, σ) is given by attaching ρ_{T_t}, ρ_{T_s}, and L_*^P to the root ρ_T. Since L_t^P, L_s^P, and L_*^P are pairwise disjoint, the tree (T, σ) is well-defined.

We now show that (T, σ) is of Type (II) by verifying that $\sigma(L_t^P) = \{r, s\}$ and $\sigma(L_s^P) = \{r, t\}$. It is easy to see that $\hat{z} \in L_{s,t}^P$ and $\hat{y} \in L_{t,s}^P$ and thus, $s \in \sigma(L_t^P)$ and $t \in \sigma(L_s^P)$. Since $\hat{z} \in L_s^P$, we can apply Property (B4.b) to conclude that $\hat{x}_2 \in N_r(\hat{z}) \subset L_s^P$. Hence, $r \in \sigma(L_s^P)$. Applying (B3.b), one similarly shows $\hat{x}_1 \in L_t^P$ and thus, $r \in \sigma(L_t^P)$. By construction, $t \notin \sigma(L_t^P)$ and $s \notin \sigma(L_s^P)$. Thus $\sigma(L_t^P) = \{r, s\}$ and $\sigma(L_s^P) = \{r, t\}$ and hence, (T, σ) is of Type (II).

It remains to show that $G(T, \sigma) = (G, \sigma)$. To this end, we put $L_1 \coloneqq L_t^P$ and $L_2 \coloneqq L_s^P$ as well as $v_1 \coloneqq \rho_{T_t}$ and $v_2 \coloneqq \rho_{T_s}$. Therefore $L_1 = L(T(v_1))$ and $\sigma(L_1) = \{r, s\}$ as well as $L_2 = L(T(v_2))$ and $\sigma(L_1) = \{r, t\}$.

In order to show $G(T, \sigma) = (G, \sigma)$, we first consider the adjacencies between vertices $L[s]$ and $L[t]$. By Conditions (B3.a) and (B3.b), we have $yz \in E(G)$ for any $y \in L[s]$, $z \in L[t]$. The same is true for $G(T, \sigma)$ by Lemma 5.18(iii). Thus the edges between vertices of color s and t in $G(T, \sigma)$ and (G, σ) coincide.

Next, we show that the neighborhood w.r.t. r of any vertex of color s and t, respectively, coincide in (G, σ) and $G(T, \sigma)$. For each $y \in L_1$ with $\sigma(y) = s$, we have $\mathrm{lca}(y, x) \prec_T \mathrm{lca}(y, x')$ for any $x \in L(T(v_1))$, $x' \notin L(T(v_1))$ with $\sigma(x) = \sigma(x') = r$. Therefore $N_r(y) \subset L_1$ in $G(T, \sigma)$ for all $y \in L_1 \cap L[s]$. By Condition (B3.b), we also have $N_r(y) \subset L_1$ in (G, σ) for all $y \in L_1 \cap L[s]$. Clearly, for any $x \in L(T(v_1))$, we have $xy \in E(G(T, \sigma))$ if and only if $\mathsf{par}(x) =$

$\mathsf{par}(y)$. Moreover we constructed (T,σ) such that $\mathsf{par}(x) = \mathsf{par}(y)$ if and only if $xy \in E(G)$. Hence, the neighborhoods $N_r(y)$ in $G(T,\sigma)$ and (G,σ) coincide for all $y \in L_1 \cap L[s]$. By similar arguments and application of (B4.b) one can show that the neighborhoods $N_r(z)$ in $G(T,\sigma)$ and (G,σ) coincide for all $z \in L_2 \cap L[t]$.

Now suppose that $y \in L[s]$ is not contained in L_1, thus $y \in L_*^P$, and let $x \in L[r]$. By construction of (T,σ), we have $\mathrm{lca}(x,y) = \rho_T$ and thus, $\mathrm{lca}(x,y) \preceq_T \mathrm{lca}(x,y')$ for any y' of color s if and only if $x \in L_2 \cup L_*^P$, hence $N_r(y) = (L_2 \cup L_*^P) \cap L[r]$ in $G(T,\sigma)$. By Condition (B3.a), we have $N_r(y) = (L_2 \cup L_*^P) \cap L[r]$ in (G,σ) as well. Hence, the neighborhoods $N_r(y)$ coincide in $G(T,\sigma)$ and (G,σ) for all $y \in L_*^P$. Similar arguments and application of (B4.a) shows that the neighborhoods $N_r(z)$ in $G(T,\sigma)$ and (G,σ) coincide for all $z \in L_*^P$.

So far, we have shown that the neighborhoods $N(y)$ and $N(z)$ of all $y \in L[s]$, resp., $z \in L[t]$ are the same in both, $G(T,\sigma)$ and (G,σ). It remains to show that this is also true for vertices $x \in L[r]$. Since $y \in N(x) \cap L[s]$ if and only if $x \in N(y) \cap L[r]$ and the $N(y)$ neighborhoods for all $y \in L[s]$ coincide in both graphs, we can conclude that $N_s(x)$ w.r.t. $G(T,\sigma)$ coincides with $N_s(x)$ w.r.t. (G,σ). The same is true for the $N_t(x)$ neighborhoods. Hence, the $N(x)$ neighborhoods in $G(T,\sigma)$ and (G,σ), resp., are identical. In summary, we have shown that $G(T,\sigma) = (G,\sigma)$, i.e., (T,σ) explains (G,σ). Hence, (G,σ) is a 3-RBMG.

Now let (G,σ) be a 3-RBMG of Type (B). By Lemma 5.23, (G,σ) is B-like w.r.t. $\langle \hat{x}_1 \hat{y} \hat{z} \hat{x}_2 \rangle$ for some $\hat{x}_1, \hat{x}_2 \in L[r]$, $\hat{y} \in L[s]$, $\hat{z} \in L[t]$, which proves (B1). Moreover, again by Lemma 5.23, the tree (T,σ) that explains (G,σ) can be chosen in a way that it is of Type (II*) and satisfies $L_1 = L(T(v_1))$, $L_2 = L(T(v_2))$ for $v_1, v_2 \in \mathsf{child}(\rho_T) \setminus L$, and $L_*^P = \mathsf{child}(\rho_T) \cap L$. Now careful application of Lemma 5.18(i)-(iv), which is left to the reader, easily shows that Conditions (B2.a) to (B4.b) are satisfied.

□

Note that some conditions in Lemma 5.24 are redundant. For instance, (B4.a) and (B4.b) are a consequence of (B2.a)-(B3.b). They are convenient, however, to describe the structure of Type (B) 3-RBMGs since they emphasize the symmetric structure of the conditions and somewhat simplify the arguments. A non-redundant set of conditions will be given in Thm. 5.6 at the end of this section.
As a direct consequence of the previous result and Lemma 5.4, we obtain the following:

Corollary 5.8. *Any (not necessarily* S*-thin) Type* (B) *3-RBMG* (G,σ) *contains an induced path* $\langle xyzx' \rangle$ *with* $\sigma(x) = \sigma(x') = r$, $\sigma(y) = s$, *and* $\sigma(z) = t$ *for distinct colors* r, s, t *such that* $N_r(y) \cap N_r(z) = \emptyset$.

We give here an alternative receipt to reconstruct a 3-RBMG (G,σ) of Type (B) that is B-like w.r.t. some P as in Def. 5.12, based on its induced subgraphs $(G_*,\sigma_*) := (G[L_*^P], \sigma_{|L_*^P})$, $(G_1,\sigma_1) := (G[L_t^P], \sigma_{|L_t^P})$, and $(G_2,\sigma_2) := (G[L_s^P], \sigma_{|L_s^P})$. This particular reconstruction and the knowledge about the structure of Type (B) RBMGs may be potentially useful for orthology detection, more precisely for the identification of false positive and

false negative orthology assignments (see Chapter 6). By (B2.a) and (B2.b), (G_1, σ_1) and (G_2, σ_2) are both disjoint unions of an arbitrary number of K_1s and K_2s. By Lemma 5.23, there exists a tree of Type (II*) that explains (G, σ) and satisfies $L(T(v_1)) = L_t^P$, $L(T(v_2)) = L_s^P$ for $v_1, v_2 \in \mathsf{child}(\rho_T) \setminus L$, and $L_*^P = \mathsf{child}(\rho_T) \cap L$. Hence, by Lemma 5.18, (G, σ) can be obtained by inserting all edges ab with

(i) $a \in L_t^P$, $b \in L_s^P$, and $\sigma(a), \sigma(b) \in \sigma(L_t^P) \triangle \sigma(L_s^P)$, and

(ii) $a \in L_i^P$, $b \in L_*^P$, and $\sigma(b) \notin \sigma(L_i^P)$ for $i \in \{s, t\}$

into the disjoint union of (G_1, σ_1), (G_2, σ_2), and (G_*, σ_*).

However, the assignment of leaves to one of the sets L_t^P, L_s^P, or L_*^P strongly depends on the choice of the corresponding induced P_4. We refer to Fig. 27 (Section 5.6) for an example. The 3-RBMG (G, σ) contains the induced P_4s $\langle a_1 b_1 c_1 a_2 \rangle$ and $\langle a_1 c_2 b_2 a_2 \rangle$, where $a_1, a_2 \in L[r]$, $b_1, b_2 \in L[s]$ and $c_1, c_2 \in L[t]$. If L_t^P, L_s^P, or L_*^P are defined w.r.t. $P = \langle a_1 b_1 c_1 a_2 \rangle$, then one obtains $L_t^P = \{a_1, b_1\}$, $L_s^P = \{a_2, c_1\}$, and $L_*^P = \{b_2, c_2\}$, from which one constructs the tree (T_1, σ). On the other hand, if $P = \langle a_1 c_2 b_2 a_2 \rangle$ is chosen as the corresponding P_4, it yields $L_t^P = \{a_1, c_2\}$, $L_s^P = \{a_2, b_2\}$, $L_*^P = \{b_1, c_1\}$, and the tree (T_2, σ).

We will return to the induced P_4s with endpoints of the same color in Section 5.6 below. We shall see that they fall into two distinct classes, which we call *good* and *bad*. All good P_4s in (G, σ) imply the same vertex sets L_t^P, L_s^P, and L_*^P. In contrast, different bad P_4s results in different vertex sets.

5.5.5 *Characterization of Type* (C) *3-RBMGs*

We continue with a characterization of Type (C) 3-RBMGs. To this end, the construction of Type (B) 3-RBMGs can be extended to a similar characterization of Type (C) 3-RBMGs. Similarly to the last subsection we start by introducing *C-like* graphs:

Definition 5.13. *Let (G, σ) be an undirected, connected, properly colored,* S*-thin graph. Moreover, assume that (G, σ) contains the hexagon $H := \langle \hat{x}_1 \hat{y}_1 \hat{z}_1 \hat{x}_2 \hat{y}_2 \hat{z}_2 \rangle$ such that $\sigma(\hat{x}_1) = \sigma(\hat{x}_2) = r$, $\sigma(\hat{y}_1) = \sigma(\hat{y}_2) = s$, and $\sigma(\hat{z}_1) = \sigma(\hat{z}_2) = t$. Then (G, σ) is* C-like w.r.t. *H if there is a vertex $v \in \{\hat{x}_1, \hat{y}_1, \hat{z}_1, \hat{x}_2, \hat{y}_2, \hat{z}_2\}$ such that $|N_c(v)| > 1$ for some color $c \neq \sigma(v)$. Suppose that (G, σ) is C-like w.r.t. $H = \langle \hat{x}_1 \hat{y}_1 \hat{z}_1 \hat{x}_2 \hat{y}_2 \hat{z}_2 \rangle$ and assume w.l.o.g. that $v = \hat{x}_1$ and $c = t$, i.e., $|N_t(\hat{x}_1)| > 1$. Then we define the following sets:*

$$
\begin{aligned}
L_t^H &:= \{x \mid \langle x \hat{z}_2 \hat{y}_2 \rangle \in \mathcal{P}_3\} \cup \{y \mid \langle y \hat{z}_1 \hat{x}_2 \rangle \in \mathcal{P}_3\} \\
L_s^H &:= \{x \mid \langle x \hat{y}_2 \hat{z}_2 \rangle \in \mathcal{P}_3\} \cup \{z \mid \langle z \hat{y}_1 \hat{x}_1 \in \rangle \mathcal{P}_3\} \\
L_r^H &:= \{y \mid \langle y \hat{x}_2 \hat{z}_1 \rangle \in \mathcal{P}_3\} \cup \{z \mid \langle z \hat{x}_1 \hat{y}_1 \rangle \in \mathcal{P}_3\} \\
L_*^H &:= V(G) \setminus (L_r^H \cup L_s^H \cup L_t^H).
\end{aligned}
$$

Again, there is a close connection between these vertex sets and trees of Type (III).

Lemma 5.25. *Let (G,σ) be an* S*-thin 3-RBMG of Type* (C) *with $|L|>6$ and color set $S=\{r,s,t\}$. Then, up to permutation of colors, (G,σ) is C-like w.r.t. the hexagon $H=\langle \hat{x}_1\hat{y}_1\hat{z}_1\hat{x}_2\hat{y}_2\hat{z}_2\rangle$ for some $\hat{x}_i\in L[r]$, $\hat{y}_i\in L[s]$, $\hat{z}_i\in L[t]$ and there exists a tree (T,σ) of Type* (III) *explaining (G,σ) such that*

(i) $L_t^H=L(T(v_1))$, $L_s^H=L(T(v_2))$, *and* $L_r^H=L(T(v_3))$ *where* $v_1,v_2,v_3\in \mathsf{child}(\rho_T)$, *and*

(ii) $L_*^H=\mathsf{child}(\rho_T)\cap L$.

Proof. We argue along the lines of the proof of Lemma 5.23. Let (G,σ) be a 3-RBMG of Type (C). Then Lemma 5.17 implies that there exists a tree (T,σ) with root ρ_T explaining (G,σ) that is of Type (III), thus in particular there are vertices $v_1,v_2,v_3\in\mathsf{child}(\rho_T)$ with $\sigma(L(T(v_1)))=\{r,s\}$, $\sigma(L(T(v_2)))=\{r,t\}$, and $\sigma(L(T(v_3)))=\{s,t\}$, and $\mathsf{child}(\rho_T)\setminus\{v_1,v_2,v_3\}\subset L$. Similar argumentation as in the proof of Thm. 5.4 (Claim 3) shows that there are leaves $\hat{x}_1,\hat{y}_1\prec_T v_1$, $\hat{x}_2,\hat{z}_1\prec_T v_2$, and $\hat{y}_2,\hat{z}_2\prec_T v_3$, where $\hat{x}_1,\hat{x}_2\in L[r]$, $\hat{y}_1,\hat{y}_2\in L[s]$, $\hat{z}_1,\hat{z}_2\in L[t]$, such that $\langle\hat{x}_1\hat{y}_1\hat{z}_1\hat{x}_2\hat{y}_2\hat{z}_2\rangle$ is a hexagon. Since $|L|>6$, there exists an additional leaf z'. W.l.o.g. we can assume that this additional vertex has color $\sigma(z')=t$. Since (T,σ) is of Type (III), there are three mutually exclusive cases: $z'\prec_T v_2$ or $z'\prec_T v_3$ or z' is incident to the root ρ_T.

Suppose first that $z'\prec_T v_2$. Lemma 5.18(ii) implies $z'\hat{y}_1\in E(G)$. Since in addition $\hat{z}_1\hat{y}_1\in E(G)$, we can conclude $|N_t(\hat{y}_1)|>1$. Similarly, if $z'\prec_T v_3$, then Lemma 5.18(ii) implies $z'\hat{x}_1\in E(G)$ and thus, as $\hat{z}_2\hat{x}_1\in E(G)$, we have $|N_t(\hat{x}_1)|>1$. Finally, if z' is incident to the root ρ_T, then Lemma 5.18(iv) implies $z'\hat{x}_1\in E(G)$ and we again obtain $|N_t(\hat{x}_1)|>1$. In summary, if $|L|>6$ and (G,σ) is of Type (C), then there is always a hexagon H and a vertex v in H such that $|N_c(v)|>1$ for some color $c\neq\sigma(v)$. Therefore (G,σ) is C-like w.r.t. some hexagon in (G,σ).

It remains to show Properties (i) and (ii). Since (G,σ) is C-like w.r.t. some hexagon H in (G,σ) and one can always shift the vertex labels along $H=\langle\hat{x}_1\hat{y}_1\hat{z}_1\hat{x}_2\hat{y}_2\hat{z}_2\rangle$ as well as permuting the colors in (G,σ), we can w.l.o.g. assume $v=\hat{x}_1$ and $c=t$, i.e., $|N_t(\hat{x}_1)|>1$. Let $x\in L[r]$ and assume first $x\in L(T(v_1))$. We show that this implies $x\in L_t^H$. Lemma 5.18(ii) implies $x\hat{z}_2\in E(G)$ and $x\hat{y}_2\notin E(G)$. Since $\hat{y}_2\hat{z}_2\in E(G)$ by definition of H, we can conclude $\langle x\hat{z}_2\hat{y}_2\rangle\in\mathcal{P}_3$. Thus $x\in L_t^H$. Hence, we have $L(T(v_1))\cap L[r]\subseteq L_t^H\cap L[r]$. Now let $x\in L_t^H$, i.e., $\langle x\hat{z}_2\hat{y}_2\rangle$ forms an induced P_3. Since $x\hat{y}_2\notin E(G)$, Lemma 5.18(ii) implies $x\notin L(T(v_2))$. In addition, $x\hat{y}_2\notin E(G)$ and Lemma 5.18(iv) imply that x cannot be incident to the root ρ_T. Moreover, $x\notin L(T(v_3))$ because $r\notin\sigma(L(T(v_3)))$ by construction of (T,σ). Hence, x must be contained in $L(T(v_1))$. Therefore we have $L(T(v_1))\cap L[r]\supseteq L_t^H\cap L[r]$, which implies $L(T(v_1))\cap L[r]=L_t^H\cap L[r]$.

Analogously, one shows $L(T(v_1))\cap L[s]=L_t^H\cap L[s]$, from which it can be inferred that $L(T(v_1))=L_t^H$. By symmetry, one similarly obtains $L(T(v_2))=L_s^H$ and $L(T(v_3))=L_r^H$, which finally shows Property (i). Property (ii) is a direct consequence of Property (i) because $L_*^H=L\setminus(L_t^H\cup L_s^H\cup L_r^H)=L\setminus(L(T(v_1))\cup L(T(v_2))\cup L(T(v_3)))=\mathsf{child}(\rho_T)\cap L$.

□

Lemma 5.26. *Let (G,σ) be an undirected, connected, S-thin, and properly 3-colored graph with color set $S=\{r,s,t\}$ and let $x\in L[r]$, $y\in L[s]$, and $z\in L[t]$. Then (G,σ) is a 3-RBMG of Type* (C) *if and only if (G,σ) is either a hexagon or $|L|>6$ and, up to permutation of colors, the following conditions are satisfied:*

(C1) *(G,σ) is C-like w.r.t. the hexagon $H=\langle\hat{x}_1\hat{y}_1\hat{z}_1\hat{x}_2\hat{y}_2\hat{z}_2\rangle$ for some $\hat{x}_i\in L[r]$, $\hat{y}_i\in L[s]$, $\hat{z}_i\in L[t]$ with $|N_t(\hat{x}_1)|>1$,*

(C2.a) *If $x\in L_*^H$, then $N(x)=L_r^H\cup(L_*^H\setminus\{x\})$,*

(C2.b) *If $x\in L_t^H$, then $N_s(x)\subset L_t^H$ and $|N_s(x)|\leq 1$, and $N_t(x)=L_*^H[t]\cup L_r^H[t]$,*

(C2.c) *If $x\in L_s^H$, then $N_t(x)\subset L_s^H$ and $|N_t(x)|\leq 1$, and $N_s(x)=L_*^H[s]\cup L_r^H[s]$*

(C3.a) *If $y\in L_*^H$, then $N(y)=L_s^H\cup(L_*^H\setminus\{y\})$,*

(C3.b) *If $y\in L_t^H$, then $N_r(y)\subset L_t^H$ and $|N_r(y)|\leq 1$, and $N_t(y)=L_*^H[t]\cup L_s^H[t]$,*

(C3.c) *If $y\in L_r^H$, then $N_t(y)\subset L_r^H$ and $|N_t(y)|\leq 1$, and $N_r(y)=L_*^H[r]\cup L_s^H[r]$,*

(C4.a) *If $z\in L_*^H$, then $N(z)=L_t^H\cup(L_*^H\setminus\{z\})$,*

(C4.b) *If $z\in L_s^H$, then $N_r(z)\subset L_s^H$ and $|N_r(z)|\leq 1$, and $N_s(z)=L_*^H[s]\cup L_t^H[s]$,*

(C4.c) *If $z\in L_r^H$, then $N_s(z)\subset L_r^H$ and $|N_s(z)|\leq 1$, and $N_r(z)=L_*^H[r]\cup L_t^H[r]$.*

In particular, L_t^H, L_s^H, L_r^H, and L_*^H are pairwise disjoint and $\hat{x}_1,\hat{y}_1\in L_t^H$, $\hat{x}_2,\hat{z}_1\in L_s^H$, $\hat{y}_2,\hat{z}_2\in L_r^H$.

Proof. If $|L|\leq 6$, it follows from Thm. 5.4 that (G,σ) is a 3-RBMG of Type (C) if and only if $|L|=6$ and (G,σ) is a hexagon. Hence, we assume that $|L|>6$ and (G,σ) satisfies conditions (C1) - (C4.c). As a consequence of (C1), if (G,σ) is an RBMG, then it must be of Type (C). In order to prove that (G,σ) is indeed an RBMG, we construct a tree (T,σ) based on the sets L_t^H, L_s^H, L_r^H, and L_*^H and show that (T,σ) explains (G,σ).

To this end, we first show that the sets L_t^H, L_s^H, L_r^H, and L_*^H are pairwise disjoint. By definition, L_*^H is disjoint from L_t^H, L_s^H, and L_r^H. Now, let $x\in L[r]$ and assume that $x\in L_t^H$. Hence, $\langle x\hat{z}_2\hat{y}_2\rangle\in\mathcal{P}_3$ which in particular implies $x\hat{z}_2\in E(G)$. Therefore $\langle x\hat{y}_2\hat{z}_2\rangle\notin\mathcal{P}_3$ and thus, $x\notin L_s^H$. Repeated analogous argumentation shows that L_t^H, L_s^H, and L_r^H are pairwise disjoint.

Moreover, for the construction of (T,σ), we show that $G[L_i^H]$ is the disjoint union of an arbitrary number of K_1s and K_2s with $i\in\{r,s,t\}$. By definition of L_t^H, we have $\sigma(L_t^H)\subseteq\{r,s\}$. Since $N_s(x)\subset L_t^H$ and $|N_s(x)|\leq 1$ for any $x\in L_t^H$ by (C2.b) as well as $N_r(y)\subset L_t^H$ and $|N_r(y)|\leq 1$ for any $y\in L_t^H$ by (C3.b), any vertex of L_t^H has at most one neighbor in L_t^H. Thus $G[L_t^H]$ is

the disjoint union of an arbitrary number of K_1s and K_2s. Similar arguments and application of Properties (C2.c) and (C4.b), resp., (C3.c) and (C4.c), show that $G[L_s^H]$, resp., $G[L_r^H]$, is the disjoint union of an arbitrary number of K_1s and K_2s.

We are now in the position to construct a tree (T, σ) based on the sets L_t^H, L_s^H, L_r^H, and L_*^H. First, for $i \in \{r, s, t\}$, we construct a caterpillar (T_i, σ_i), with root ρ_{T_i}, on the leaf set L_i^H such that $\mathsf{par}(a) = \mathsf{par}(b)$ for any $a, b \in L_i^H$ with $\sigma(a) \neq \sigma(b)$ if and only if $ab \in E(G)$. Since $G[L_i^H]$ is the disjoint union of an arbitrary number of K_1s and K_2s, the tree T_i is well-defined. It is, however, not unique as the order of inner vertices in T_i is arbitrary. Then (T, σ) is given by attaching ρ_{T_t}, ρ_{T_s}, ρ_{T_r}, and L_*^H to the root ρ_T. Since L_t^H, L_s^H, L_r^H, and L_*^H are pairwise disjoint, the tree (T, σ) is well-defined. It is easy to verify that $\hat{x}_1, \hat{y}_1 \in L_t^H$, $\hat{x}_2, \hat{z}_1 \in L_s^H$, and $\hat{y}_2, \hat{z}_2 \in L_r^H$. Therefore $\sigma(L_t^H) = \{r, s\}$, $\sigma(L_s^H) = \{r, t\}$, and $\sigma(L_r^H) = \{s, t\}$. This implies that (T, σ) is of Type (III). To this end, let $v_1 := \rho_{T_t}$, $v_2 := \rho_{T_s}$ and $v_3 := \rho_{T_r}$ and thus, $L(T(v_1)) = L_t^H$, $L(T(v_2)) = L_s^H$, and $L(T(v_3)) = L_r^H$, respectively.

It remains to show that $G(T, \sigma) = (G, \sigma)$. We first consider the adjacencies of the vertices with color r. Let $x \in L[r]$. Suppose first $x \in \mathsf{child}(\rho_T)$, i.e., $x \in L_*^H$ in (G, σ). Clearly, any leaf (with color different from $\sigma(x)$) that is incident to the root ρ_T is a neighbor of x in $G(T, \sigma)$, i.e., $L_*^H \setminus \{x\} \subseteq N(x)$. Moreover, since there is no leaf of color r in $L(T(v_3))$, we have $L_r^H \subseteq N(x)$ in $G(T, \sigma)$ by Lemma 5.18(iv). Hence, $L_r^H \cup (L_*^H \setminus \{x\}) \subseteq N(x)$ in $G(T, \sigma)$. Furthermore, since r is contained in $\sigma(L(T(v_1)))$ as well as in $\sigma(L(T(v_2)))$, we can apply Lemma 5.18(iv) to conclude that x is not adjacent to any vertex in $L(T(v_1)) = L_t^H$ and $L(T(v_2)) = L_s^H$ in $G(T, \sigma)$. Thus $N(x) \subseteq L_r^H \cup (L_*^H \setminus \{x\})$ and therefore, $N(x) = L_r^H \cup (L_*^H \setminus \{x\})$ in $G(T, \sigma)$ for all $x \in L[r] \cap L_*^H$. By Property (C2.a), the latter is also satisfied in (G, σ) for all $x \in L[r] \cap L_*^H$. Hence, the respective neighborhoods of all $x \in L[r] \cap L_*^H$ in $G(T, \sigma)$ and (G, σ) coincide.

Now let $x \in L(T(v_1)) = L_t^H$. By construction and Lemma 5.18(i), we have $xy \in E(G(T, \sigma))$, resp., $xy \in E(G)$ for $y \in L[s]$ if and only if $\mathsf{par}(x) = \mathsf{par}(y)$. Hence, the respective neighborhoods $N_s(x)$ of all $x \in L(T(v_1)) \cap L[r] = L_t^H \cap L[r]$ in $G(T, \sigma)$ and (G, σ) coincide. Now consider the neighborhood $N_t(x)$ in $G(T, \sigma)$. Since $r \in \sigma(L(T(v_2))) = L_s^H$, Lemma 5.18(ii) implies that x is not adjacent to any vertex in L_s^H. Hence, as $t \notin \sigma(L_t^H)$, if follows $N_t(x) \subseteq L_*^H[t] \cup L_r^H[t]$. Since there is no leaf of color t in $L(T(v_1))$, we have $L_*^H[t] \subseteq N_t(x)$ by Lemma 5.18(iv). Moreover, as $r \notin \sigma(L(T(v_3)))$, Lemma 5.18(ii) implies $L_r^H[t] = L(T(v_3)) \cap L[t] \subseteq N_t(x)$. Hence, $L_*^H[t] \cup L_r^H[t] \subseteq N_t(x)$ and we therefore conclude $N_t(x) = L_*^H[t] \cup L_r^H[t]$. By Property (C2.b), the latter is also satisfied in (G, σ) for all $x \in L(T(v_1)) = L_t^H$. Hence, the respective neighborhoods $N_t(x)$ of all $x \in L_t^H \cap L[r]$ are identical in $G(T, \sigma)$ and (G, σ). Since $N_t(x) \cup N_s(x) = N(x)$, the neighborhoods $N(x)$ coincide in $G(T, \sigma)$ and (G, σ) for every $x \in L_t^H \cap L[r]$. By similar arguments, one can show that the same is true for any $x \in L_s^H \cap L[r]$.

By symmetry, analogous arguments show that the neighborhoods of leaves with color s or t are the same in (G, σ) and $G(T, \sigma)$. We therefore conclude $(G, \sigma) = G(T, \sigma)$, i.e., (T, σ) explains (G, σ).

Conversely, let (G, σ) be a connected, S-thin 3-RBMG of Type (C). By Thm. 5.4, (G, σ) is either a hexagon, or $|L| > 6$ and (G, σ) contains a hexagon H of the form (r, s, t, r, s, t). In the latter case, Lemma 5.25 implies that (G, σ) is always C-like w.r.t. some hexagon $H = \langle \hat{x}_1 \hat{y}_1 \hat{z}_1 \hat{x}_2 \hat{y}_2 \hat{z}_2 \rangle$ with $\hat{x}_i \in L[r]$, $\hat{y}_i \in L[s]$, $\hat{z}_i \in L[t]$. Similar arguments as in the proof of Lemma 5.25 show that w.l.o.g. we can assume $|N_t(\hat{x}_1)| > 1$. Hence, (G, σ) satisfies Property (C1). Moreover, Lemma 5.25 implies that there exists a tree (T, σ) of Type (III) that explains (G, σ) and such that $L(T(v_1)) = L_t^H$, $L(T(v_2)) = L_s^H$, $L(T(v_3)) = L_r^H$, and $L_*^H = \mathsf{child}(\rho_T) \cap L$. Now, careful application of Lemma 5.18(i)-(iv), which is left to the reader, shows that Conditions (C2.a) to (C4.c) are satisfied. □

Together with Lemma 5.4, the latter result immediately implies:

Corollary 5.9. *Any (not necessarily* S*-thin) Type* (C) *3-RBMG* (G, σ) *contains a hexagon* $\langle xyzx'y'z' \rangle$ *with* $\sigma(x) = \sigma(x') = r$, $\sigma(y) = \sigma(y') = s$, *and* $\sigma(z) = \sigma(z') = t$ *for distinct colors* r, s, t *such that* $|N_c(v)| > 1$ *for some* $v \in \{x, x', y, y', z, z'\}$ *and* $c \neq \sigma(v)$.

If (G, σ) is a 3-RBMG of Type (C), an analogous construction as in the case of Type (B) 3-RBMGs can be used to obtain (G, σ) from the sets L_t^H, L_s^H, L_r^H, and L_*^H. Again, this information is useful for correcting the orthology graph.

If $|L| = 6$, then (G, σ) is already a hexagon $H = \langle x_1 y_1 z_1 x_2 y_2 z_2 \rangle$ such that, up to permutation of the colors, $\sigma(x_i) = r$, $\sigma(y_i) = s$, and $\sigma(z_i) = t$, $i \in \{1, 2\}$. This 3-colored graph is explained by the two distinct trees $T_1 := ((x_1, y_1), (z_1, x_2), (y_2, z_2))$ and $T_2 := ((y_1, z_1), (x_2, y_2), (z_2, x_1))$, given in standard Newick tree format. These two trees induce different leaf sets $L(T(v_i))$, where $v_i \in \mathsf{child}(\rho_T) \cap V^0(T)$ in the corresponding tree. One can show, however, that for $|L| > 6$, every hexagon defines the same sets L_i^H, $i \in \{t, s, r\}$, and L_*^H. To this end, we will need the following technical result:

Lemma 5.27. *Let* (T, σ) *be a tree of Type* (III) *with root* ρ_T *explaining a connected,* S*-thin 3-RBMG* (G, σ) *and let* $H := \langle \hat{x}_1 \hat{y}_1 \hat{z}_1 \hat{x}_2 \hat{y}_2 \hat{z}_2 \rangle$ *be a hexagon in* (G, σ) *such that* $\hat{x}_i \in L[r]$, $\hat{y}_i \in L[s]$, *and* $\hat{z}_i \in L[t]$ *for distinct colors* r, s, t *and* $1 \leq i \leq 2$. *Then* $\hat{x}_i, \hat{y}_i, \hat{z}_i \notin \mathsf{child}(\rho_T)$, $1 \leq i \leq 2$.

Proof. By definition of (T, σ), there exist distinct $v_1, v_2, v_3 \in \mathsf{child}(\rho_T)$ such that $\sigma(L(T(v_1))) = \{r, s\}$, $\sigma(L(T(v_2))) = \{r, t\}$, $\sigma(L(T(v_3))) = \{s, t\}$, and $\mathsf{child}(\rho_T) \setminus \{v_1, v_2, v_3\} \subset L$. Assume, for contradiction, that $\hat{x}_1 \in \mathsf{child}(\rho_T)$. Then either $\hat{y}_1 \in \mathsf{child}(\rho_T)$ or, by Lemma 5.18(iv), $\hat{y}_1 \preceq_T v_3$. In the latter case, Lemma 5.18(i) implies $\hat{z}_1 \preceq_T v_3$ and thus, $\hat{x}_1 \hat{z}_1 \in E(G)$ by Lemma 5.18(iv); contradicting that H is a hexagon. Hence, $\hat{x}_1 \notin \mathsf{child}(\rho_T)$. Due to symmetry, we can apply similar arguments to the remaining vertices $\hat{x}_2, \hat{y}_i, \hat{z}_i$, $1 \leq i \leq 2$, to show that none of them is contained in $\mathsf{child}(\rho_T)$. □

We are now in the position to prove the uniqueness of L_i^H, $i \in \{r, s, t\}$, and L_*^H.

Lemma 5.28. *Let* (G, σ) *be a connected,* S*-thin 3-RBMG of Type* (C) *with leaf set* $|L| > 6$. *Moreover, let the sets* L_t^H, L_s^H, L_r^H, *and* L_*^H *be defined w.r.t. an induced hexagon* $H := \langle \hat{x}_1 \hat{y}_1 \hat{z}_1 \hat{x}_2 \hat{y}_2 \hat{z}_2 \rangle$ *with* $|N_t(\hat{x}_1)| > 1$, *where* $\hat{x}_i \in L[r]$, $\hat{y}_i \in L[s]$ *and* $\hat{z}_i \in L[t]$ *for distinct colors* r, s, t. *Then, for any hexagon* H'

of the form (r,s,t,r,s,t)*, we have* $L_t^{H'} = L_t^H$*,* $L_s^{H'} = L_s^H$*,* $L_r^{H'} = L_r^H$*, and* $L_*^{H'} = L_*^H$*.*

Proof. Let (T,σ) be a leaf-colored tree explaining (G,σ), which, by Thm. 5.4 can be chosen to be of Type (III), i.e., there are distinct $v_1,v_2,v_3 \in \mathsf{child}(\rho_T)$ such that $\sigma(L(T(v_1))) = \{r,s\}$, $\sigma(L(T(v_2))) = \{r,t\}$, $\sigma(L(T(v_3))) = \{s,t\}$, and $\mathsf{child}(\rho_T) \setminus \{v_1,v_2,v_3\} \subset L$. In particular, Lemma 5.25 implies that (T,σ) can be chosen such that $L_t^H = L(T(v_1))$, $L_s^H = L(T(v_2))$, $L_r^H = L(T(v_3))$, and $L_*^H = \mathsf{child}(\rho_T) \cap L$. Lemma 5.27 implies $V(H) \cap \mathsf{child}(\rho_T) = \emptyset$. Since $\hat{x}_1$ has more than one neighbor of color t, Lemma 5.18(i) and S-thinness of (G,σ) imply that $\hat{x}_1$ cannot be contained in $L(T(v_2))$ as otherwise $|N_t(\hat{x}_1)| \leq 1$. Hence, $\hat{x}_1 \preceq v_1$. Applying Lemma 5.18(i)+(ii), we then conclude that $\hat{y}_1 \preceq_T v_1$, $\hat{x}_2,\hat{z}_1 \preceq_T v_2$, and $\hat{y}_2,\hat{z}_2 \preceq_T v_3$. In other words, $\hat{x}_1,\hat{y}_1 \in L_t^H$, $\hat{x}_2,\hat{z}_1 \in L_s^H$, and $\hat{y}_2,\hat{z}_2 \in L_r^H$.

We proceed to show that the leaf sets L_t^H, L_s^H, L_r^H, and L_*^H remain unchanged if they are taken w.r.t. some other vertex $v \in V(H) \setminus \{\hat{x}_1\}$ with $|N_c(v)| > 1$ for some color $c \neq \sigma(v)$. Note that, as (G,σ) is explained by (T,σ) and $\hat{x}_1 \in L_t^H$, we can apply Property (C2.b) to conclude $|N_s(\hat{x}_1)| \leq 1$. Suppose first $v = \hat{y}_1$ and $|N_c(\hat{y}_1)| > 1$. Since $\hat{y}_1 \in L_t^H$, we can apply Property (C3.b) and obtain $|N_r(\hat{y}_1)| \leq 1$. Hence, we have $c = t$. The definition of L_t with $v = \hat{y}_1$ and $c = t$ implies that $L_t^{H'} = \{y \mid \langle y\hat{z}_1\hat{x}_2\rangle \in \mathcal{P}_3\} \cup \{x \mid \langle x\hat{z}_2\hat{y}_2\rangle \in \mathcal{P}_3\} = L_t^H$. Similarly, one obtains $L_s^{H'} = L_s^H$ and $L_r^{H'} = L_r^H$. Now let $v = \hat{z}_1$ and $|N_c(\hat{z}_1)| > 1$. Then as (T,σ) explains (G,σ) and $\hat{z}_1 \in L_s^H$, Property (C4.b) implies $|N_r(\hat{z}_1)| \leq 1$. Hence, $c = s$. Again, the definition of L_t^H with $v = \hat{z}_1$ and $c = s$ implies that $L_t^{H'} = L_t^H$. Similarly, the definition of L_s^H and L_r^H with $v = \hat{z}_1$ and $c = s$ shows that $L_s^{H'} = L_s^H$, and $L_r^{H'} = L_r^H$. Applying similar arguments to $v = \hat{y}_2$, $v = \hat{z}_2$, and $v = \hat{x}_2$ under the assumption that $|N_c(v)| > 1$ for some color $c \neq \sigma(v)$, shows that v and c always induce the same leaf sets L_t^H, L_s^H, L_r^H. The latter implies that also the set L_*^H is independent from the particular choice of the vertices v in H.

Now let $H' := \langle x_1y_1z_1x_2y_2z_2\rangle \neq H$ with $x_i \in L[r]$, $y_i \in L[s]$, $z_i \in L[t]$. Lemma 5.27 implies that x_1 and x_2 are not incident to the root of (T,σ), hence $x_1,x_2 \in L(T(v_1)) \cup L(T(v_2))$. Assume, for contradiction, that they are contained in the same subtree, say $x_1,x_2 \in L(T(v_1))$. Then, as $x_2z_1 \in E(G)$ and $\sigma(L(T(v_1))) = \{r,s\}$, Lemma 5.18(ii) implies that z_1 cannot reside within a subtree that contains leaves of color r, thus $z_1 \in L(T(v_3))$. Therefore we can again apply Lemma 5.18(ii) to conclude that $x_1z_1 \in E(G)$; a contradiction since H' is a hexagon. Analogously one shows that x_1 and x_2 cannot be both located in the subtree $T(v_2)$. Hence, we can w.l.o.g. assume $x_1 \in L(T(v_1))$. Then, by construction of (T,σ), we have $\mathrm{lca}_T(x_1,z) = \mathrm{lca}_T(\hat{x}_1,z)$ for any $z \in L[t]$, thus $N_t(x_1) = N_t(\hat{x}_1)$ and in particular $|N_t(x_1)| > 1$. Applying Lemma 5.27 and analogous argumentation as for H yields $x_1,y_1 \preceq_T v_1$, $x_2,z_1 \preceq_T v_2$, and $y_2,z_2 \preceq_T v_3$. Thus, in other words, $x_1,y_1 \in L_t^H$, $x_2,z_1 \in L_s^H$, and $y_2,z_2 \in L_r^H$.

Consider first L_t^H and let $x \in L[r]$. By definition, $x \in L_t^H$ if and only if $\langle x\hat{z}_2\hat{y}_2\rangle$ is an induced P_3. Since $\sigma(L(T(v_1))) = \{r,s\}$ and $\sigma(L(T(v_3))) = \{s,t\}$, Lemma 5.18(ii) implies $\langle xz_2y_2\rangle \in \mathcal{P}_3$, i.e., $x \in L_t^{H'}$. Conversely, suppose $x \in L_t^{H'}$, thus $\langle xz_2y_2\rangle \in \mathcal{P}_3$. Since (T,σ) explains (G,σ) and $y_2,z_2 \preceq_T v_3$, Lemma 5.18(ii)+(iv) implies $x \in L(T(v_1)) = L_t^H$. Hence, $L_t^H \cap L[r] = L_t^{H'} \cap$

$L[r]$. Similar arguments show $L_t^H \cap L[s] = L_t^{H'} \cap L[s]$ and thus, $L_t^H = L_t^{H'}$. By symmetry, analogous arguments yield $L_s^H = L_s^{H'}$ and $L_r^H = L_r^{H'}$. Taken together, this implies $L_*^H = L_*^{H'}$. Analogous argumentation as used for H shows that any vertex $v \in V(H')$ with $|N_c(v)| > 1$ with $c \neq \sigma(v)$, induces the same leaf sets $L_t^{H'}$, $L_s^{H'}$, $L_r^{H'}$, and $L_*^{H'}$, which finally completes the proof. □

In contrast to Observation 5.4 for Type (B) 3-RBMGs, we thus obtain the following uniqueness result.

Corollary 5.10. *Let (G, σ) be a connected S-thin 3-RBMG of Type* (C)*. If (G, σ) is C-like w.r.t. a hexagon H, then (G, σ) is C-like w.r.t. every hexagon of the form (r, s, t, r, s, t).*

5.5.6 *Characterization of 3-RBMGs and Algorithmic Results*

For later reference, finally, we summarize the main results of this section, i.e., Thm. 5.4 and the characterizations of the three types in Lemmas 5.21, 5.24, and 5.26:

Theorem 5.6. *An undirected, connected, properly 3-colored, S-thin graph (G, σ) is a 3-RBMG if and only if it satisfies either conditions* (A1) *and* (A2)*,* (B1)-(B3.b)*, or* (C1)-(C3.c) *and thus, is of Type* (A)*,* (B)*, or* (C)*.*

Proof. By Thm. 5.4, any S-thin connected 3-RBMG (G, σ) must be either of Type (A), (B), or (C). Lemma 5.21 implies that (G, σ) is a 3-RBMG of Type (A) if and only if it satisfies (A1) and (A2). By Lemma 5.24, (G, σ) is a 3-RBMG of Type (B) if and only if Properties (B1)-(B4.b) are satisfied. However, as the neighborhoods of all vertices of one color can clearly be recovered from the neighborhoods of all vertices of different color, Properties (B4.a) and (B4.b) are redundant, i.e., (G, σ) is a Type (B) 3-RBMG if and only if (B1) to (B3.b) are satisfied. One analogously argues that Properties (C4.a)-(C4.c) are redundant. □

Let us now consider the question how difficult it is to decide whether a given graph is a 3-RBMG or not. It easy to see that all conditions in Thm. 5.6 can be tested in polynomial time. In case (G, σ) is a 3-RBMG, we are also interested in a tree that can explain (G, σ). Unless (G, σ) is of Type (A), we have to construct the leaf sets L_s^P, L_t^P, L_*^P, or L_r^H, L_s^H, L_t^H, L_*^H, respectively. Instead of checking each of the conditions for Type (B) or Type (C) graphs in Thm. 5.6, we can construct the tree (T, σ) directly from the sets L_i^X, $i \in \{r, s, t\}$, $X \in \{P, H\}$ (cf. Lemma 5.23, resp., 5.25) and test whether or not (T, σ) explains (G, σ). The overall structure of this algorithm is summarized in Algorithm 4. We first show in Lemma 5.29 that Algorithm 4 indeed recognizes 3-RBMGs and, in the positive case, returns a tree. The proof of Lemma 5.29 provides at the same time a description of the single steps of Algorithm 4. We then continue to show in Lemma 5.30 that Algorithm 4 runs in $O(|V(G/\mathsf{S})|^2|E(G/\mathsf{S})| + |E(G)|)$ time for a given input graph (G, σ).

Algorithm 4 3-RBMG Recognition and Construction of Tree

Require: Properly 3-colored, connected graph (G', σ').

1: $(G, \sigma) \leftarrow (G'/\mathsf{S}, \sigma'_{/\mathsf{S}})$
2: **if** `Test_Type_A`(G, σ) = `true` **then**
3: $(T, \sigma) \leftarrow$ `Build-Tree`(G, σ)
4: **goto** Line 18
5: **else**
6: Find one hexagon H of the form (r, s, t, r, s, t)
7: **if** (G, σ) is C-like w.r.t. H **then**
8: compute L_r^H, L_s^H, L_t^H, L_*^H
9: $(T, \sigma) \leftarrow$ `Build-Tree`$((G, \sigma), L_r^H, L_s^H, L_t^H, L_*^H)$ ▷ *cf. Lemma 5.25*
10: **if** (T, σ) explains (G, σ) **then**
11: **goto** Line 18
12: **else if** (G, σ) satisfies Def. 5.12(i) for some $P = \langle xyzx' \rangle \in \mathcal{P}_4$ with $\sigma(x) = \sigma(x')$ **then**
13: compute L_s^P, L_t^P, L_*^P
14: $(T, \sigma) \leftarrow$ `Build-Tree`$((G, \sigma), L_s^P, L_t^P, L_*^P)$ ▷ *cf. Lemma 5.23*
15: **if** (T, σ) explains (G, σ) **then**
16: **goto** Line 18
17: **return** "(G, σ) is not a 3-RBMG"
18: construct final tree (T', σ') for (G', σ') based on (T, σ)
19: **return** (T, σ) and (T', σ')

Lemma 5.29. *Algorithm 4 determines if a given properly 3-colored, connected graph (G', σ') is a 3-RBMG and, in the positive case, returns a tree (T', σ') that explains (G', σ')*

Proof. Given a properly 3-colored, connected graph (G', σ'), we first compute $(G, \sigma) = (G'/\mathsf{S}, \sigma'_{/\mathsf{S}})$. By construction, (G, σ) remains properly 3-colored and, by Lemma 5.4, (G, σ) is S-thin and connected.

In Line 2, if (G, σ) is of Type (A), then we can compute the tree (T, σ) that explains (G, σ) as constructed for the "if-direction" in the proof of Lemma 5.21, and jump to Line 18.

If (G, σ) is not of Type (A), then we proceed by testing if (G, σ) is of Type (C). To this end, we search first for one hexagon H of the form (r, s, t, r, s, t) in Line 6. If such a hexagon H exists, we check if (G, σ) is C-like w.r.t. H. By Cor. 5.10, it is indeed sufficient to test C-likeness for one hexagon only. If (G, σ) is C-like w.r.t. H, then we compute the sets L_r^H, L_s^H, L_t^H, L_*^H (Line 8). We proceed in Line 9 to construct a tree (T, σ) based on the set L_r^H, L_s^H, L_t^H, L_*^H according to Lemma 5.25. Now, to test if (G, σ) is of Type (C), we can again apply Lemma 5.25 which implies that it suffices show that (T, σ) explains (G, σ). If this is the case, we again jump to Line 18 and, if not, proceed to check if (G, σ) is of Type (B).

If (G, σ) is neither of Type (A) nor (C), then either (G, σ) is not a 3-RBMG or it must be of Type (B). Thus we continue in Line 12-15 to test if (G, σ) can be explained by some tree (T, σ). To this end, Observation 5.4 implies that we must check for every $P \in \mathcal{P}_4$ (for which the two endpoints have the same color),

whether (G, σ) satisfies Def. 5.12(i). If this is not the case for any such induced P_4, then Lemma 5.23 implies that (G, σ) is not of Type (B). Together with the preceding tests, we can conclude that (G, σ) is not a 3-RBMG. Hence, the algorithm stops in Line 17 and returns "(G, σ) is not a 3-RBMG". Otherwise, if (G, σ) satisfies Def. 5.12(i) w.r.t. P, we construct a tree (T, σ) based on the set L_s^P, L_t^P, L_*^P according to Lemma 5.23. Again by Lemma 5.23, it is now sufficient to show that (T, σ) explains (G, σ) in order to test if (G, σ) is a 3-RBMG. Since the preceding tests already have established that (G, σ) is neither of Type (A) nor (C), we can conclude that (G, σ) is of Type (B). If (G, σ) is a 3-RBMG, then we jump to Line 18, otherwise we stop again in Line 17 and the algorithm returns "(G, σ) is not a 3-RBMG".

Finally, after having verified that (G, σ) is indeed a 3-RBMG and constructed (T, σ), the algorithm reaches Line 18. Lemma 5.6 implies that (G', σ') is a 3-RBMG. Moreover, the construction in the last part of the proof of Lemma 5.6 shows how to obtain a tree (T', σ') that explains (G', σ') from (T, σ). In Line 19, the respective trees (T', σ') and (T, σ) are returned. □

Lemma 5.30. *Let (G', σ') is an undirected, properly 3-colored, connected graph and let $n = |V(G/\mathsf{S})|$, $m = |E(G/\mathsf{S})|$, and $m' = |E(G')|$. Algorithm 4 processes (G', σ') in $O(mn^2 + m')$ time.*

Proof. In a worst case, Observation 5.4 implies that we need to list all induced P_4s $\langle \hat{x}_1 \hat{y} \hat{z} \hat{x}_2 \rangle$ with $\sigma(\hat{x}_1) = \sigma(\hat{x}_2)$. Since for any edge yz in G, there exist at most $(n-2)(n-3)$ possible combinations of vertices x and x' such that $\langle xyzx' \rangle$ forms an induced P_4, there are at most $O(mn^2)$ such paths. Hence, the global runtime of the algorithm cannot be better than $O(mn^2)$. Thus only rough upper bounds for all other subtask are provided to show that they stay within $O(mn^2)$ time.

The computation of the relation S, its equivalence classes and $(G = (V, E), \sigma) = (G'/\mathsf{S}, \sigma'_{/\mathsf{S}})$ in Line 1 can be done in a similar fashion as outlined by Hammack et al. [87, Section 24.4] in $O(|E(G')|)$ time (cf. [87, Lemma 24.10]).

To test whether (G, σ) is of Type (A), we first apply Cor. 5.7 and check for which colors $i \in \{r, s, t\}$ we have $|L[i]| = 1$, which can be done in $O(n)$ time. We then apply Lemma 5.21 and verify if $G \notin \mathcal{P}_3$. Note that the latter task can be done in constant time since we can check if $n = 3$ and, in the positive case, if the three vertices of G are pairwisely connected by an edge in constant time $O(3)$. We apply Lemma 5.21 again, and check for all colors $i \in \{r, s, t\}$ with $L[i] = \{x\}$, if x is a hub-vertex and if $|N(y)| < 3$ for every $y \in V \setminus \{x\}$. Both of the latter tasks can be done in $O(n)$ time. If such a color and vertex exists, then (G, σ) is of Type (A) and we can build the tree (T, σ) that explains (G, σ). To this end, we apply the construction as in the "if-direction" of the proof of Lemma 5.21. We first construct the caterpillar $(T_2, \sigma_{|L_2})$ with leaf set $L_2 = \{y \mid y \neq x, |N(y)| = 2\}$ and root ρ_{T_2}. It is easy to see that L_2 can be constructed in $O(n)$ time. For the tree T_2 we add vertices such that $\mathsf{par}(y) = \mathsf{par}(z)$ for any $y, z \in L_2$ with $\sigma(y) \neq \sigma(z)$ if and only if $yz \in E(G)$. Clearly, this task can be done in $O(m)$ time. To construct the final tree (T, σ) we need to check if $|V \setminus L_2| = 2$ or $|V \setminus L_2| = 3$, which can be done trivially in

$O(n^2)$ time. All remaining steps to construct (T, σ) can be done in constant time. Hence, for the construction of (T, σ) we need $O(m + n^2) = O(n^2)$ time. In summary, Line 2 and 3 have overall time complexity $O(n^2)$.

We continue by testing if (G, σ) is of Type (C) in Line 6-10. In Line 6, we first check if (G, σ) is C-like w.r.t. some hexagon H. Note that all candidate hexagons must be of the form (r, s, t, r, s, t). In order to find such hexagons, we first compute the pairwise distances between all vertices in $O(n^3) \subseteq O(n^2 m)$ time (Floyd-Warshall [36]). Then we fix one of the colors, say r. Clearly, two vertices in $L[r]$ that have distance larger or smaller than 3, cannot be both located on such a hexagon. Thus, for all vertices $x, x' \in L[r]$ with distance $d(x, x') = 3$, we proceed as follows: We check for all edges yz with $y \in L[s]$, $z \in L[t]$, if $x \in N_r(y)$, $x' \in N_r(z)$, $x' \notin N_r(y)$, $x \notin N_r(z)$. If this is the case, $\langle xyzx' \rangle \in \mathcal{P}_4$ and we store $\langle xyzx' \rangle$ in the list $P_{(x,x')}[s, t]$. Similarly, if for the edge yz with $y \in L[s]$, $z \in L[t]$ we have $x \in N_r(z)$, $x' \in N_r(y)$, $x' \notin N_r(z)$, $x \notin N_r(y)$, then we put $\langle xzyx' \rangle$ in the list $P_{(x,x')}[t, s]$. For each edge the latter tests can be done in constant time, e.g. by using the adjacency matrix representation of (G, σ). As soon as we have found two vertices $x, x' \in L[r]$ such that each list $P_{(x,x')}[s, t]$ and $P_{(x,x')}[t, s]$ contains at least one element $\langle xyzx' \rangle$ and $\langle xz'y'x' \rangle$ such that yz' and zy' do not form an edge, we have found a hexagon $H = \langle xyzx'y'z' \rangle$ of the form (r, s, t, r, s, t). Thus, for a given pair $x, x' \in L[r]$, finding a hexagon that contains x and x' can be done in $O(m)$ time. As the latter may be repeated for all $x, x' \in L[r]$, we can conclude that finding a hexagon of the form (r, s, t, r, s, t) in Line 6, can be done $O(|L[r]|^2 m) = O(n^2 m)$ time. Clearly, the test if (G, σ) is C-like w.r.t. H in Line 7 can be done in constant time. Now the sets L_r^H, L_s^H, L_t^H, L_*^H are computed in Line 8. To determine these sets, we compute for each edge uv in H all vertices $w \in V \setminus (L[\sigma(u)] \cup L[\sigma(v)])$ such that $\langle wuv \rangle \in \mathcal{P}_3$. The latter can be done in $O(n)$ for each edge in H. Since H has only a constant number of edges, all sets L_r^H, L_s^H, L_t^H can be constructed in $O(n)$ time. The set L_*^H can then be trivially constructed in $O(n^2)$ time. Now we continue in Line 9 to construct a tree (T, σ) as in Lemma 5.25. Similar arguments as in the Type (A) case show that (T, σ) can be constructed in $O(m)$ time. Finally, we check in Line 10 if (T, σ) explains (G, σ). To this end, we note that T has $O(n)$ vertices. Moreover it was shown in [198], that the last common ancestor of x and y can be accessed in constant time after an $O(|V(T)|) = O(n)$ time preprocessing step. Hence, for each edge $xy \in E(G)$, we check if $\text{lca}(x, y) \preceq_T \text{lca}(x, y')$ and $\text{lca}(x, y) \preceq_T \text{lca}(x', y)$ for all $x' \in L[\sigma(x)]$ and $y' \in L[\sigma(y)]$ in $O(n^2)$. As this has to repeated for all edges of G, Line 10 takes $O(mn^2)$ time. In summary, testing if (G, σ) is of Type (C) in Line 6-10 can be done in $O(mn^2)$ time.

In Line 12, we verify if (G, σ) satisfies Def. 5.12(i) w.r.t. some $P \in \mathcal{P}_4$. Note that there are at most mn^2 P_4s $\langle abcd \rangle$ with $\sigma(a) = \sigma(d)$ in (G, σ). Listing all such induced P_4s can therefore trivially be done $O(mn^2)$ time by reusing the list $P_{(x,x')}[s, t]$ from the Type (C) case. For each induced P_4 $\langle abcd \rangle$ with $\sigma(a) = \sigma(d)$ we can verify the condition in Def. 5.12(i) in at most $O(n)$ time. Thus Line 12 requires $O(mn^2)$ time.

In Line 13, we need to construct the sets L_s^P, L_t^P, and L_*^P. Assume that $P = \langle \hat{x}_1 \hat{y} \hat{z} \hat{x}_2 \rangle$ is of the form (r, s, t, r). To construct the set $L_{t,s}^P$, we have

$y \in L^P_{t,s}$ if the edge $y\hat{z}$ and every $x \in N_r(y)$ form an induced P_3. For each edge $y\hat{z}$ the latter can be tested in $O(n)$ time. To obtain $L^P_{t,s}$ the latter test must be repeated for all edges $y\hat{z}$ with $y \in L[s]$. Thus $L^P_{t,s}$ can be constructed in $O(mn)$ time. The set $L^P_{t,r}$ is the disjoint union of two sets L' and L'', where the first set L' contains all $x \in L[r]$ for which $x, y, \hat{z}$ induce a P_3 with $N_r(y) = \{x\}$ and the second set L'' contains all $x \in L[r]$ with $N_s(x) = \emptyset$ whenever $L[s] \setminus L^P_{t,s} \neq \emptyset$. By similar arguments as for $L^P_{t,s}$, the set L' can be constructed in $O(mn)$ time. For the set L'', observe that $L[s] \setminus L^P_{t,s} \neq \emptyset$ can be trivially verified in at most $O(n^2)$ time and $N_s(x) = \emptyset$ can be verified in $O(n)$ time for a given $x \in L[r]$. To obtain L'' we must repeat the latter for all $x \in L[r]$ and hence, end up with a time complexity $O(n^3) \subseteq O(n^2m)$. In summary, the set $L^P_{t,s}$ and $L^P_{t,r}$ can be constructed in $O(mn)$ and $O(n^2m)$ time, respectively. Therefore L^P_t can be constructed in $O(n^2m)$ time. By symmetry, the construction of L^P_s can be done in $O(n^2m)$ time as well. The set $L^P_* = V \setminus (L^P_t \cup L^P_s)$ can then trivially be constructed in $O(n^2)$ time. Now we continue in Line 14 to construct a tree (T, σ) as in Lemma 5.23. Similar arguments as in the Type (A) case show that (T, σ) can be constructed in $O(m)$ time. Finally, we check in Line 15 if (T, σ) explains (G, σ). By similar arguments as in the Type (C) case, the latter task can be done in $O(mn^2)$ time. In summary, Lines 12-15 require $O(mn^2)$ time.

Finally we construct, in Line 18 the tree (T', σ') for (G', σ') based on the tree (T, σ). Given the equivalence classes as computed in the first step (Line 1), one can construct (T', σ') as in the last part of the proof of Lemma 5.6. Thus, for each of the n leaves x, we can check in $O(n)$ time in which class it is contained and then expand the leaf x by $|[x]|$ vertices. As there are at most $O(n')$ vertices that we may additionally add to (T, σ), we can construct (T', σ') in $O(n + n') = O(n') \subseteq O(m')$ time. Since the task of computing the quotient graph (G, σ) already takes $O(m')$ time, we end up with an overall runtime of $O(mn^2 + m')$.

□

5.6 THE GOOD, THE BAD, AND THE UGLY: INDUCED P_4S

In order to gain a better understanding of Type (B) 3-RBMGs, we consider here in more detail the influence of the choice of the "reference" P_4 on the definition of the vertex sets L^P_s, L^P_t, and L^P_* that determine the structure of (G, σ). Those P_4s can be classified as so-called *good*, *bad*, and *ugly* quartets. Quartets will play an essential role for the characterization of 3-RBMGs as we shall see later. In particular, the sets L^P_s, L^P_t, and L^P_* can be determined by good quartets and are independent of the choice of the respective good quartet. Moreover, a we will see in Chapter 6, good quartets play an important role for the detection of false positive and false negative orthology assignments.

Observation 5.5. *An n-RBMG does not contain an induced P_4 on two colors. Moreover, any induced P_4 with three distinct colors is either of the Type $\langle xyzx' \rangle$ or $\langle xyx'z \rangle$ with $\sigma(x) = \sigma(x')$.*

Proof. As shown by Cor. 5.1, there is no induced P_4 with only two colors since all 2-RBMGs are complete bipartite graphs. Hence, if we have three

distinct colors, then exactly two vertices have the same color. Since RBMGs are properly colored, these vertices cannot be adjacent, leaving only the two alternatives $\langle xyzx' \rangle$ and $\langle xyx'z \rangle$.

□

Nevertheless, an RBMG on more than three colors may also contain induced P_4s with four distinct colors. Consider, for instance, the tree $((a_1, b_1, c), (a_2, b_2, d))$, given in Newick format, where $\sigma(a_i) = A$, $\sigma(b_i) = B$, $\sigma(c) = C$, and $\sigma(d) = D$, $i \in \{1, 2\}$, and A, B, C, and D are pairwise distinct colors. Then the RBMG $G(T, \sigma)$ contains the 4-colored induced P_4 $\langle a_1 c d b_2 \rangle$. A characterization for n-RBMGs that are cographs will be given later in Thm. 5.8. For now we will restrict our attention to P_4s with three colors only.

Definition 5.14 (good, bad, and ugly quartets[1])**.** *Let $(\vec{G}, \sigma)$ be a BMG with symmetric part (G, σ) and let $Q := \{x, x', y, z\} \subseteq L$ with $x, x' \in L[r]$, $y \in L[s]$, and $z \in L[t]$. The set Q, resp., the induced subgraph $(\vec{G}[Q], \sigma_{|Q})$ is*

- *a* good quartet *if (i) $\langle xyzx' \rangle$ is an induced P_4 in (G, σ) and (ii) $(x, z), (x', y) \in E(\vec{G})$ and $(z, x), (y, x') \notin E(\vec{G})$,*
- *a* bad quartet *if (i) $\langle xyzx' \rangle$ is an induced P_4 in (G, σ) and (ii) $(z, x), (y, x') \in E(\vec{G})$ and $(x, z), (x', y) \notin E(\vec{G})$,*
- *an* ugly quartet *if $\langle xyx'z \rangle$ is an induced P_4 in (G, σ).*

Fig. 27 shows an example of an RBMG containing a good quartet. Note that good, bad, and ugly quartets cannot appear in RBMGs whose induced 3-colored subgraphs are all Type (A) 3-RBMGs: By definition, these do not contain induced P_4s.
The next result shows that any induced P_4 is a quartet of one of those three types:

Lemma 5.31. *Let (G, σ) be an RBMG, Q a set of four vertices with three colors, $G[Q] \in \mathcal{P}_4$, and $(\vec{G}, \sigma)$ a BMG containing (G, σ). Then Q is either a good, a bad, or an ugly quartet.*

Proof. By Obs. 5.5, any induced P_4 is either of the form $\langle xyx'z \rangle$ or $\langle xyzx' \rangle$. In the first case Q is an ugly quartet. For the remainder of the proof we thus assume $\langle xyzx' \rangle$, and w.l.o.g. suppose that the vertex colors are $\sigma(x) = \sigma(x') = r$, $\sigma(y) = s$, and $\sigma(z) = t$.

Let (T, σ) be a leaf-colored tree that explains $(\vec{G}, \sigma)$, and thus, by assumption, also (G, σ). Since $\langle xyzx' \rangle$ is an induced P_4 in (G, σ), the edge xz cannot be contained in $E(G)$. Hence, we are left with three cases: (i) $(x, z) \in E(\vec{G})$ and $(z, x) \notin E(\vec{G})$, (ii) $(z, x) \in E(\vec{G})$ and $(x, z) \notin E(\vec{G})$, and (iii) $(x, z), (z, x) \notin E(\vec{G})$.
Case (i). We have $x' \in N_r^+(z)$ and $x \notin N_r^+(z)$, i.e., $\mathrm{lca}_T(x', z) \prec_T \mathrm{lca}_T(x, z) =: u$. This implies $\mathrm{lca}_T(x, x') = u$. Moreover, $(x, y) \in E(G)$ implies $\mathrm{lca}_T(x, y) \preceq_T \mathrm{lca}_T(x', y)$. In case of equality, we have $\mathrm{lca}_T(x, y) = \mathrm{lca}_T(x', y) \succeq_T u$. Thus

1 Best enjoyed with proper soundtrack at https://www.youtube.com/watch?v=XjehlT1VjiU

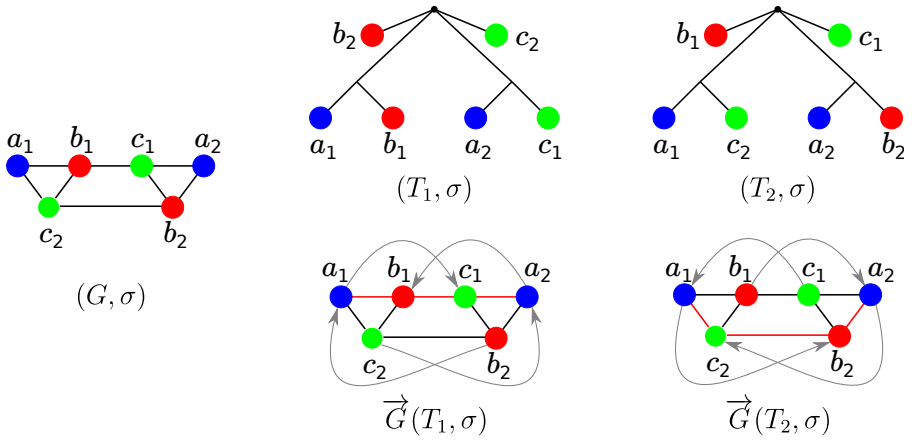


Fig. 27. The S-thin 3-RBMG (G, σ) is explained by two trees (T_1, σ) and (T_2, σ) that induce distinct BMGs $\vec{G}(T_1, \sigma)$ and $\vec{G}(T_2, \sigma)$. In $\vec{G}(T_1, \sigma)$, $P^1 = \langle a_1 b_1 c_1 a_2 \rangle$ defines a good quartet, while $P^2 = \langle a_1 c_2 b_2 a_2 \rangle$ induces a bad quartet. In $\vec{G}(T_2, \sigma)$ the situation is reversed. Moreover, the quartets for P^1 and P^2 induce different leaf sets. Denoting the leaf colors "blue", "red", and "green" by r, s, and t, respectively, we obtain $L_t^{P^1} = \{a_1, b_1\}$, $L_s^{P^1} = \{a_2, c_1\}$, and $L_*^{P^1} = \{b_2, c_2\}$, while $L_s^{P^2} = \{a_1, c_2\}$, $L_t^{P^2} = \{a_2, b_2\}$, and $L_*^{P^2} = \{b_1.c_1\}$. The good quartets in $\vec{G}(T_1, \sigma)$ and $\vec{G}(T_2, \sigma)$ are indicated by red edges. The induced paths $\langle a_1 b_1 c_1 b_2 \rangle$ and $\langle a_2 c_1 b_1 c_2 \rangle$ are examples of ugly quartets.

$x \in N_r^+(y)$ implies $x' \in N_r^+(y)$. Hence, since $x'y \notin E(G)$, there must exist a leaf $y' \in L[s]$ such that $\mathrm{lca}_T(x', y') \prec_T \mathrm{lca}_T(x', y)$. Then $\mathrm{lca}_T(x', z) \prec_T u$ implies either $\mathrm{lca}_T(z, y') \prec_T u \preceq_T \mathrm{lca}_T(x', y)$ or $\mathrm{lca}_T(z, y') = \mathrm{lca}_T(x', y') \prec_T \mathrm{lca}_T(x', y)$. Either alternative contradicts $yz \in E(G)$. Therefore $\mathrm{lca}_T(x, y) \prec_T \mathrm{lca}_T(x', y)$. Together with $\mathrm{lca}_T(x, x') = u$ this implies $\mathrm{lca}_T(x, y) \prec_T u$. Now let $u_y \succeq_T \mathrm{lca}_T(x, y)$ and $u_z \succeq_T \mathrm{lca}_T(x', z)$, where $u_y, u_z \in \mathsf{child}(u)$. Then $yz \in E(G)$ implies $t \notin \sigma(L(T(u_y)))$ and $s \notin \sigma(L(T(u_z)))$. Hence, $(x, z), (x', y) \in E(\vec{G})$ and $(z, x), (y, x') \notin E(\vec{G})$, i.e., $\{x, y, z, x'\}$ forms a good quartet.

Case (ii). We have $x, x' \in N_r^+(z)$, thus $u := \mathrm{lca}_T(x, z) = \mathrm{lca}_T(x', z)$. On the other hand, $(x, z) \notin E(\vec{G})$ implies that there exists a leaf $z' \in L[t]$ such that $\mathrm{lca}_T(x, z') \prec_T \mathrm{lca}_T(x, z)$. Hence, as $z \in N_t^+(x')$, we have distinct $u_x, u_{x'}, u_z \in \mathsf{child}(u)$ such that $x, z' \prec_T u_x$, $x' \preceq_T u_{x'}$, and $z \preceq_T u_z$. Moreover, $yz \in E(G)$ implies $\mathrm{lca}_T(y, z) \preceq_T \mathrm{lca}_T(y, z')$, thus we either have (a) $\mathrm{lca}_T(y, z) \prec_T \mathrm{lca}_T(y, z')$ or (b) $\mathrm{lca}_T(y, z) = \mathrm{lca}_T(y, z')$. In both cases, we have $u_x \prec_T \mathrm{lca}(x, y)$, thus, since $xy \in E(G)$, it follows $s \notin \sigma(L(T(u_x)))$. This in particular implies $\mathrm{lca}_T(x', y) \preceq_T \mathrm{lca}_T(x, y)$. Similarly, since $zx' \in E(G)$, we have $r \notin \sigma(L(T(u_z)))$ and $t \notin \sigma(L(T(u_{x'})))$. Moreover, as $x'y \notin E(G)$ and $\mathrm{lca}_T(x', y) \preceq_T \mathrm{lca}_T(x, y)$, there must exist a leaf $y' \in L[s]$ with $\mathrm{lca}_T(x', y') \prec_T \mathrm{lca}_T(x', y)$. In Case (a), we have $y \in L(T(v_z))$ and thus $\mathrm{lca}_T(x', y) = u$, which implies $y' \preceq_T u_{x'}$. In summary, this implies for Case (a) $x, z' \prec_T u_x$, $x', y' \prec_T u_{x'}$, and $y, z \prec_T u_z$ as well as $\sigma(L(T(u_x))) = \{r, t\}$, $\sigma(L(T(u_{x'}))) = \{r, s\}$, and $\sigma(L(T(u_z))) = \{s, t\}$. Hence, $(z, x), (y, x') \in E(\vec{G})$ and $(x, z), (x', y) \notin E(\vec{G})$, i.e., $\{x, y, z, x'\}$ is a bad quartet in $(\vec{G}, \sigma)$. In Case (b), if $\mathrm{lca}_T(x', y') \succeq u$, then $\mathrm{lca}_T(x, y') = \mathrm{lca}_T(x', y') \prec_T \mathrm{lca}_T(x', y)$, contradicting $xy \in E(G)$. Hence, $y' \preceq_T u_{x'}$. This implies $\mathrm{lca}_T(x, y) = u$ since

otherwise $\mathrm{lca}_T(x,y') = u \prec_T \mathrm{lca}_T(x,y)$; again a contradiction to $xy \in E(G)$. Let $u_y \in \mathsf{child}(u)$ be such that $y \preceq_T u_y$. Since $xy, yz \in E(G)$, we conclude $\sigma(L(T(u_y))) = \{s\}$. Moreover, $yz \in E(G)$ then implies $s \notin \sigma(L(T(u_z)))$. Summarizing Case (b), we thus have $x, z' \prec_T u_x$, $x', y' \prec_T u_{x'}$, $y \preceq_T u_y$, and $z \prec_T u_z$ as well as $\sigma(L(T(u_x))) = \{r,t\}$, $\sigma(L(T(u_{x'}))) = \{r,s\}$, $\sigma(L(T(u_y))) = \{s\}$, and $\sigma(L(T(u_z))) = \{t\}$. One now easily checks that $\{x,y,z,x'\}$ again forms a bad quartet in $(\vec{G},\sigma)$.
Case (iii). Let $u := \mathrm{lca}_T(x,x')$. Then $(x,z),(z,x) \notin E(\vec{G})$ implies $\mathrm{lca}_T(x',z) \prec_T \mathrm{lca}_T(x,z)$ and there must exist some leaf $z' \in L[t]$ such that $\mathrm{lca}_T(x,z') \prec_T \mathrm{lca}_T(x,z)$. Hence, there are $u_x, u_{x'} \in \mathsf{child}(u)$ with $\mathrm{lca}_T(x,z') \preceq_T u_x$ and $\mathrm{lca}_T(x',z) \preceq_T u_{x'}$. By construction, we therefore have either $\mathrm{lca}_T(x,y) \prec_T \mathrm{lca}_T(x',y)$ or $\mathrm{lca}_T(x,y) = \mathrm{lca}_T(x',y) \succeq_T u$. The first case implies $\mathrm{lca}_T(y,z') \prec_T u = \mathrm{lca}_T(y,z)$, which contradicts $yz \in E(G)$. Hence, it must hold $\mathrm{lca}_T(x,y) = \mathrm{lca}_T(x',y) \succeq_T u$ and thus, $x' \in N_r^+(y)$ because $x \in N_r^+(y)$. Consequently, since $x'y \notin E(G)$, there must be some $y' \in L[s]$ such that $\mathrm{lca}_T(x',y') \prec_T \mathrm{lca}_T(x',y)$. The same argumentation as in Case (i) shows that $\mathrm{lca}_T(x',z) \prec_T u$ implies either $\mathrm{lca}_T(z,y') \prec_T u \preceq_T \mathrm{lca}_T(x',y)$ or $\mathrm{lca}_T(z,y') = \mathrm{lca}_T(x',y') \prec_T \mathrm{lca}_T(x',y)$, which in either case contradicts $yz \in E(G)$. We therefore conclude that Case (iii) is impossible.

□

We immediately find the following result that links good quartets to 3-RBMGs of Type (B):

Lemma 5.32. *Let (G,σ) be an undirected, connected, S-thin, and properly 3-colored graph that contains an induced path P on four vertices. If (G,σ) satisfies* (B1) *to* (B4.b) *w.r.t. P, then there exists a tree (T,σ) explaining (G,σ) such that P is a good quartet in $\vec{G}(T,\sigma)$.*

Proof. If P satisfies (B1) to (B4.b), then, by Lemma 5.24, (G,σ) is a 3-RBMG of Type (B). Thus, according to Lemma 5.23, there exists a Type (II*) tree (T,σ) with root ρ_T such that $L_t^P = L(T(v_1))$, $L_s^P = L(T(v_2))$ for distinct $v_1, v_2 \in \mathsf{child}_T(\rho_T) \setminus L$, and $L_*^P = \mathsf{child}_T(\rho_T) \cap L$, that explains (G,σ). In particular, by Property (B1), we have $P := \langle xyzx' \rangle$ with $\sigma(x) = \sigma(x') = r$, $\sigma(y) = s$ and $\sigma(z) = t$ for distinct colors r, s, and t. Now, as $x, y \in L_t^P$ and $x', z \in L_s^P$ by Lemma 5.24 and, by definition, $\sigma(L_t^P) = \{r,s\}$, $\sigma(L_s^P) = \{r,t\}$, one easily checks that P is indeed a good quartet in $\vec{G}(T,\sigma)$. □

We continue with some basic result about ugly quartets before analyzing good and bad quartets in more details.

Lemma 5.33. *Let $\langle xyx'z \rangle$ be an ugly quartet in some connected, S-thin 3-RBMG (G,σ) and (T,σ) with root ρ_T a tree of Type* (II) *or* (III) *that explains (G,σ). Then the children $v_a \in \mathsf{child}(\rho_T)$ with $a \in \{x,x',y,z\}$ and $a \preceq_T v_a$ satisfy exactly one of the following conditions:*

(i) $v_x = x$, $v_{x'} = v_z$, $v_y \neq y$, *and* v_x, v_y, v_z *are pairwise distinct,*

(ii) $v_x = v_{x'} = v_z \neq v_y$ *and* $v_y \neq y$,

(iii) $v_x \neq x$, $v_{x'} = x'$, $v_y = v_z$, *and* v_x, $v_{x'}$, v_y *are pairwise distinct,*

(iv) $v_x \neq x$, $v_{x'} = x'$, $v_y \neq y$, $v_z = z$, *and* v_x, $v_{x'}$, v_y, v_z *are pairwise distinct,*

(v) $v_x \neq x$, $v_{x'} = x'$, $v_y = y$, $v_z \neq z$, *and* v_x, $v_{x'}$, v_y, v_z *are pairwise distinct.*

In particular, x, x', *and* y *can never reside within the same subtree* $T(v_x)$. *Indeed, all cases may appear.*

Proof. Let L be the vertex set of G and r, s, t be three distinct colors in (G, σ), where $\sigma(x) = \sigma(x') = r$, $\sigma(y) = s$, and $\sigma(z) = t$. We start by considering the two cases (a) $v_x = x$, i.e., $x \in \mathsf{child}(\rho_T)$, and (b) $v_x \neq x$.

We first assume Case (a), that is, $x \in \mathsf{child}(\rho_T)$. Note first that this immediately implies $x' \notin \mathsf{child}(\rho_T)$, i.e., $v_{x'} \neq x'$, because (G, σ) is S-thin (cf. Lemma 5.17). Moreover, $xy, x'y \in E(G)$ implies $\mathrm{lca}_T(x, y) = \mathrm{lca}_T(x', y) = \rho_T$ and consequently, we have $s \notin \sigma(L(T(v_{x'})))$ because $x'y \in E(G)$. Thus, since $v_{x'} \neq x'$, Lemma 5.7 implies $t \in \sigma(L(T(v_{x'})))$. Consequently, $z \in L(T(v_{x'}))$ as otherwise, there is some $z' \in L(T(v_{x'}))$ such that $\mathrm{lca}(x', z') \prec_T \mathrm{lca}(x', z)$, which contradicts $x'z \in E(G)$. Since $xy \in E(G)$, $x \in \mathsf{child}(\rho_T)$ implies either $y \in \mathsf{child}(\rho_T)$ or $\sigma(L(T(v_y))) = \{s, t\}$ as otherwise $\mathrm{lca}(x'', y) \prec_T \mathrm{lca}(x, y)$ for some $x'' \in L(T(v_y)) \cap L[r]$, contradicting $xy \in E(G)$. If the first case is true, we have $\mathrm{lca}(y, z) \preceq_T \mathrm{lca}(y, z')$ by construction and, as $s \notin \sigma(L(T(v_{x'})))$, $\mathrm{lca}(y, z) \preceq_T \mathrm{lca}(y', z)$ for any $y' \in L[s]$ and $z' \in L[t]$, i.e., $yz \in E(G)$; a contradiction since $\langle xyx'z \rangle$ is an induced P_4. Hence, it must hold $\sigma(L(T(v_y))) = \{s, t\}$, which in particular implies $v_y \neq y$ and $v_y \neq v_{x'}, v_x$. Using Lemma 5.18, one easily checks that, if $\mathsf{par}(x') = \mathsf{par}(z)$, $\langle xyx'z \rangle$ is indeed an induced P_4 in (G, σ), which finally shows Statement (i).

Now assume that Case (b) is true, i.e., $v_x \neq x$. Then, by Lemma 5.7, the subtree $(T(v_x), \sigma_{|L(T(v_x))})$ must contain at least two colors. Assume first, for contradiction, that $s \in \sigma(L(T(v_x)))$. Then, as $xy, x'y \in E(G)$, Lemma 5.18 implies that $x' \in L(T(v_x))$ and $\mathsf{par}(x) = \mathsf{par}(x') = \mathsf{par}(y)$, which contradicts the S-thinness of (G, σ). Hence, $\sigma(L(T(v_x))) = \{r, t\}$, and in particular $v_x \neq v_y$.

If $v_x = v_{x'}$, it follows from $x'z \in E(G)$ that $z \preceq_T v_x$ (cf. Lemma 5.18), i.e., $v_x = v_z$. Moreover, since $s \notin \sigma(L(T(v_x)))$ and $yz \notin E(G)$, Lemma 5.18(iv) implies $v_y \neq y$. Hence, by Lemma 5.7, it must hold $\sigma(L(T(v_y))) = \{s, t\}$. Choosing $\mathsf{par}(x') = \mathsf{par}(z) \neq \mathsf{par}(x)$, one can again use Lemma 5.18 in order to show that $\langle xyx'z \rangle$ forms an induced P_4, which proves (ii).

On the other hand, if $v_x \neq v_{x'}$, then $xy, x'y \in E(G)$ requires $\mathrm{lca}(x, y) = \mathrm{lca}(x', y)$, hence $v_y \neq v_x, v_{x'}$. Again, $x'y \in E(G)$ then implies $s \notin \sigma(L(T(v_{x'})))$. Since $\sigma(L(T(v_{x'}))) = \sigma(L(T(v_x))) = \{r, t\}$ is not possible by construction of a tree of Type (II) or (III), $\sigma(L(T(v_{x'})))$ contains only color r, hence $v_{x'} = x'$ by Lemma 5.7. It finally remains to distinguish the two cases $v_y = v_z$ and $v_y \neq v_z$. In the first case, if $\mathsf{par}(y) \neq \mathsf{par}(z)$ in (T, σ), we can apply Lemma 5.18 to show $yz \notin E(G)$ and furthermore, that $\langle xyx'z \rangle$ is again an induced P_4. This yields Statement (iii).

If the latter case is true, i.e., $v_y \neq v_z$, then, as $xy, x'y, x'z \in E(G)$, we have in particular $r \notin \sigma(L(T(v_y))), \sigma(L(T(v_z)))$. Furthermore, since $yz \notin E(G)$, there is either some $z' \in L(T(v_y)) \cap L[t]$ such that $\mathrm{lca}(y, z') \prec_T \mathrm{lca}(y, z)$, or some $y' \in L(T(v_z)) \cap L[s]$ such that $\mathrm{lca}(y', z) \prec_T \mathrm{lca}(y, z)$. Note that, since $v_y \neq v_z$, $\sigma(L(T(v_y))) = \sigma(L(T(v_z))) = \{s, t\}$ is not possible by construction

of (T, σ). Hence, by applying Lemma 5.7 to these two cases, we either obtain (iv) or (v). Again, Lemma 5.18 easily shows that $\langle xyx'z \rangle$ is an induced P_4 in (G, σ), which concludes the proof. □

We will from now on focus on good and bad quartets only as they are of particular interest for the characterization of Type (B) 3-RBMGs. We start with some basic result.

Lemma 5.34. *Let (G, σ) be an RBMG and $P := \langle xyzx' \rangle$ an induced P_4 in (G, σ) with $\sigma(x) = \sigma(x')$, let (T, σ) be a tree explaining (G, σ), and let $v := \mathrm{lca}_T(x, x', y, z)$. Then the distinct children $v_i \in \mathsf{child}(v)$ satisfy exactly one of the three alternatives*

(i) $x, y \preceq_T v_1$ and $x', z \preceq_T v_2$,

(ii) $x \preceq_T v_1$, $y, z \preceq_T v_2$, and $x' \preceq_T v_3$,

(iii) $x \preceq_T v_1$, $y \preceq_T v_2$, $x' \preceq_T v_3$, and $z \preceq_T v_4$.

Indeed, all three cases may appear.

Proof. Let $\sigma(x) = \sigma(x') = r$, $\sigma(y) = s$, and $\sigma(z) = t$.
Suppose first $x, y \in L(T(v_1))$ for some $v_1 \in \mathsf{child}(v)$. If $z \preceq_T v_1$, then $x' \in L(T(v_1))$ since otherwise $\mathrm{lca}_T(x, z) \prec_T \mathrm{lca}_T(x', z)$, contradicting $zx' \in E(G)$. Thus $\mathrm{lca}_T(x, x', y, z) \prec_T v$; a contradiction to the definition of v. Hence, $z \notin L(T(v_1))$. Then $yz \in E(G)$ implies $t \notin \sigma(L(T(v_1)))$, hence in particular $v = \mathrm{lca}_T(x, z) \preceq_T \mathrm{lca}_T(x, z')$ for any $z' \in L[t]$, i.e., $z \in N_t^+(x)$. Since $xz \notin E(G)$, it must therefore hold $\mathrm{lca}_T(x', z) \prec_T \mathrm{lca}_T(x, z) = v$, thus $x', z \in L(T(v_2))$ for some $v_2 \in \mathsf{child}(v) \setminus \{v_1\}$. This implies Case (i).

Now suppose x and y are located in different subtrees below v, i.e., $x \preceq_T v_1$ and $y \preceq_T v_2$ for distinct children $v_1, v_2 \in \mathsf{child}(v)$. Since $xy \in E(G)$ and $\mathrm{lca}_T(x, y) = v$, we conclude that $r \notin \sigma(L(T(v_2)))$ and $s \notin \sigma(L(T(v_1)))$. This immediately implies $x' \notin L(T(v_1))$ as otherwise (a) $\mathrm{lca}_T(x, y) = \mathrm{lca}_T(x', y)$ results in $x' \in N_r^+(y)$, and (b) $\mathrm{lca}_T(x', y) \preceq_T \mathrm{lca}_T(x', y')$ for any $y' \in L[s]$, hence $x'y \in E(G)$; a contradiction. Therefore there must be a child $v_3 \neq v_1, v_2$ of v such that $x' \preceq_T v_3$. As a consequence, z cannot be contained in $L(T(v_1))$ since this would imply $\mathrm{lca}_T(x, z) \prec_T \mathrm{lca}_T(x', z)$, contradicting $x'z \in E(G)$. Suppose $z \in L(T(v_3))$. Then, since $yz \in E(G)$, we have $t \notin \sigma(L(T(v_2)))$ and $s \notin \sigma(L(T(v_3)))$. As $r \notin \sigma(L(T(v_2)))$, we conclude $\sigma(L(T(v_2))) = \{s\}$ and $\sigma(L(T(v_3))) = \{r, t\}$. Clearly, this implies $yx' \in E(G)$; a contradiction. Therefore $z \notin L(T(v_3))$, and we thus either have $z \preceq_T v_2$ or there exists another child v_4 of v ($v_4 \neq v_1, v_2, v_3$) such that $z \preceq_T v_4$. The latter shows that one of the Cases (ii) and (iii) may occur. However, we need to ensure that both can happen given the existence of the induced P_4 $\langle xyzx' \rangle$. Let us first assume $z \in L(T(v_2))$, thus $\sigma(L(T(v_2))) = \{s, t\}$. If $\sigma(L(T(v_1))) = \{r, t\}$ and $\sigma(L(T(v_3))) = \{r, s\}$, then one easily checks that $\langle xyzx' \rangle$ is an induced P_4 in (G, σ), which implies Statement (ii). On the other hand, if $z \preceq_T v_4$ and $\sigma(L(T(v_1))) = \{r, t\}$, $\sigma(L(T(v_2))) = \{s\}$, $\sigma(L(T(v_3))) = \{r, s\}$ $\sigma(L(T(v_4))) = \{t\}$, then $\langle xyzx' \rangle$ also forms an induced P_4 in (G, σ), i.e., Case (iii) is true. □

It turns out that the location of good quartets in any tree is strictly constrained:

Lemma 5.35. *Let $(\vec{G},\sigma)$ be a BMG containing a good quartet $\langle xyzx'\rangle$, (T,σ) a tree explaining $(\vec{G},\sigma)$, and $v := \mathrm{lca}(x,x',y,z)$. Then $x,y \preceq_T v_1$ and $x',z \preceq_T v_2$ for some distinct $v_1,v_2 \in \mathsf{child}(v)$.*

Proof. Let $v := \mathrm{lca}_T(x,x',y,z)$ and $v_1 \in \mathsf{child}(v)$ such that $x \preceq_T v_1$. Suppose first $y \notin L(T(v_1))$, hence in particular $\mathrm{lca}_T(y,x') \preceq_T v = \mathrm{lca}_T(y,x)$. Since $xy \in E(G(T,\sigma))$, this implies $x' \in N^+_{\sigma(x)}(y)$ in $(\vec{G},\sigma)$; a contradiction to $\langle xyzx'\rangle$ forming a good quartet. Hence, $y \preceq_T v_1$. As (T,σ) must satisfy one of the three cases of Lemma 5.34 and the only possible case is (i), we can now conclude $x,y \preceq_T v_1$ and $x',z \preceq_T v_2$. □

Note that the latter result in addition shows that the Cases (ii) and (iii) in Lemma 5.34 must correspond to bad quartets. Lemma 5.35 can now be used to show that the any good quartet in an BMG is endowed with the same coloring.

Corollary 5.11. *Let (G,σ) be a connected, S-thin 3-RBMG of Type* (B)*, $(\vec{G},\sigma)$ a BMG containing (G,σ) as symmetric part, and $Q = \langle xyzx'\rangle$ with $\sigma(x) = \sigma(x')$ a good quartet in $(\vec{G},\sigma)$. Then every good quartet with $\langle x_1y_1z_1x_1'\rangle \in \mathcal{P}_4$ has colors $\sigma(x_1) = \sigma(x_1') = \sigma(x)$, $\sigma(y_1) = \sigma(y)$, and $\sigma(z_1) = \sigma(z)$.*

Proof. Since (G,σ) a of Type (B), any leaf-colored tree (T,σ) with root ρ_T explaining (G,σ) is of Type (II) (cf. Thm. 5.4). Hence, there are distinct $v_1,v_2 \in \mathsf{child}(\rho_T)$ with $|\sigma(L(T(v_1)))| = |\sigma(L(T(v_2)))| = 2$ and $\mathsf{child}(\rho_T) \setminus \{v_1,v_2\} \subset L$. By Lemma 5.35, we have $x,y \preceq_T v_1$ and $x',z \preceq_T v_2$, hence $\sigma(L(T(v_1))) = \{\sigma(x),\sigma(y)\}$ and $\sigma(L(T(v_2))) = \{\sigma(x),\sigma(z)\}$. Therefore, the statement follows directly from Lemma 5.35.

□

We are now in the position to formulate one of the main results of this section:

Lemma 5.36. *Let (G,σ) be a connected, S-thin 3-RBMG of Type* (B) *and $(\vec{G},\sigma)$ a BMG containing (G,σ) as its symmetric part. Moreover, let L_s^Q, L_t^Q, and L_*^Q be defined w.r.t. a good quartet $Q := \langle x_1y_1z_1x_1'\rangle$, where $x_1,x_1' \in L[r]$, $y_1 \in L[s]$, and $z_1 \in L[t]$ for distinct colors r,s,t. Then, for any good quartet Q', it holds $L_s^{Q'} = L_s^Q$, $L_t^{Q'} = L_t^Q$, and $L_*^{Q'} = L_*^Q$.*

Proof. Let (T,σ) with root ρ_T be a leaf-colored tree that explains $(\vec{G},\sigma)$ and thus, also (G,σ). Since (G,σ) is of Type (B), we can choose (T,σ) to be of Type (II) by Theorem 5.4, i.e., there are distinct children $v_1,v_2 \in \mathsf{child}(\rho_T)$ with $|\sigma(L(T(v_1)))| = |\sigma(L(T(v_2)))| = 2$ such that these two subtrees have exactly one color in common, and $\mathsf{child}(\rho_T) \setminus \{v_1,v_2\} \subset L$. Applying Lemma 5.23, we can choose (T,σ) such that it is of Type (II*) and satisfies $L_t^Q = L(T(v_1))$, $L_s^Q = L(T(v_2))$, and $L_*^Q = \mathsf{child}(\rho_T) \cap L$. On the other hand, Lemma 5.35 implies $x_1,y_1 \preceq_T v_1$ and $x_1',z_1 \preceq_T v_2$.

Now let $Q' := \langle x_2y_2z_2x_2'\rangle$, $Q' \neq Q$, be another good quartet in $(\vec{G},\sigma)$. By Cor. 5.11, we have $x_2,x_2' \in L[r]$, $y_2 \in L[s]$, and $z_2 \in L[t]$. Lemma 5.35 and the structure of Type (II) trees then imply $x_2,y_2 \preceq_T v_1$ and $x_2',z_2 \preceq_T v_2$. Consider first $L_t^{Q'}$ and let $x \in L[r], y \in L[s]$. Then, by definition, $y \in L_t^{Q'}$ if and only if $\langle x'yz_2\rangle \in \mathcal{P}_3$ for each $x' \in N_r(y)$. Since any two leaves of color s and t are reciprocal best matches in (G,σ) by Lemma 5.18(iii) and $x'z_2 \notin E(G)$ for any

$x' \in N_r(y)$ only if $x' \notin L(T(v_2))$ by Lemma 5.18(i)+(iv), we have $\langle x'yz_2\rangle \in \mathcal{P}_3$ for any $x' \in N_r(y)$ if and only if $\langle x'yz_1\rangle \in \mathcal{P}_3$ for any $x' \in N_r(y)$. Hence, $y \in L_t^{Q'}$ if and only if $y \in L_t^Q$, i.e., $L_t^Q \cap L[s] = L_t^{Q'} \cap L[s]$. If $N_s(x) = \emptyset$, the latter by definition of L_t^Q immediately implies that $x \in L_t^{Q'}$ if and only if $x \in L_t^Q$. On the other hand, if $N_s(x) \neq \emptyset$, then $x \in L_t^{Q'}$ if and only if $N_r(y') = \{x\}$ for some induced P_3 $\langle xy'z_2\rangle$. This can only be true if $y' \in L(T(v_1))$ since otherwise, $y'z_2x_2'$ forms a circle, thus $|N_r(y')| > 1$. Consequently, $x \in L(T(v_1))$ by Lemma 5.18(i). Since $y'z' \in E(G)$ for any $y' \in L[s], z' \in L[t]$ by Lemma 5.18(ii) and $x'z \notin E(G)$ for any $z \in L(T(v_2)) \cap L[t]$ by Lemma 5.18(ii), we conclude that $\langle xy'z_2\rangle$ is an induced P_3 with $N_r(y') = \{x\}$ if and only if $\langle xy'z_1\rangle$ is an induced P_3 with $N_r(y') = \{x\}$, hence $L_t^Q \cap L[r] = L_t^{Q'} \cap L[r]$. We therefore conclude $L_t^Q = L_t^{Q'}$.

By symmetry, an analogous argument shows $L_s^Q = L_s^{Q'}$. Together this finally implies $L_*^Q = L_*^{Q'}$, which completes the proof. □

Fig. 27 shows that good and bad quartets do not necessarily imply the same leaf sets L_s^P, L_t^P.

The restriction of a BMG $\vec{G}(T,\sigma)$ to a subset $S' \subset S$ of colors is an induced subgraph of $\vec{G}(T,\sigma)$ explained by the restriction of (T,σ) to the leaves with colors in S' and thus, again a BMG (cf. Observation 4.2). Since $G(T,\sigma)$ is the symmetric part of $\vec{G}(T,\sigma)$, it inherits this property. In particular, we have

Observation 5.6. *If (G,σ) is an n-RBMG, $n \geq 3$, explained by (T,σ), then, for any three colors $r,s,t \in S$, the restricted tree (T_{rst},σ_{rst}) explains (G_{rst},σ_{rst}) and (G_{rst},σ_{rst}) is an induced subgraph of (G,σ).*

This observation will play an important role in the proof of the following lemma as well as in Section 5.7.

Lemma 5.37. *Let $(\vec{G},\sigma)$ be a BMG. Then the symmetric part (G,σ) contains an induced 3-colored P_4 whose endpoints have the same color if and only if $(\vec{G},\sigma)$ contains a good quartet.*

Proof. Note that, if $(\vec{G},\sigma)$ contains a good quartet, then its symmetric part (G,σ) by definition contains a 3-colored P_4 $\langle abcd\rangle$ with $\sigma(a) = \sigma(d)$.

Conversely, suppose that (G,σ) contains a 3-colored induced P_4 $\langle abcd\rangle$ whose endpoints have the same color $\sigma(a) = \sigma(d)$. Moreover, let S be the color set of (G,σ) and (T,σ) be a tree explaining $(\vec{G},\sigma)$ and thus also (G,σ). W.l.o.g. we can assume that the three colors of $\langle abcd\rangle$ are $r,s,t \in S$ without explicitly stating the particular coloring of the vertices a,b,c, and d. By definition, (G_{rst},σ_{rst}) contains the induced path $\langle abcd\rangle$. W.l.o.g. we may assume that (G_{rst},σ_{rst}) is connected; otherwise the proof works analogously for the connected component of (G_{rst},σ_{rst}) that contains $\langle abcd\rangle$.

Let us first assume that (G_{rst},σ_{rst}) is S-thin. In this case, we can apply Thm. 5.4 and conclude that (G_{rst},σ_{rst}) is either of Type (B) or (C), i.e., the restricted tree (T_{rst},σ_{rst}) with root $\rho := \mathrm{lca}_T(L[r] \cup L[s] \cup L[t])$ that explains (G_{rst},σ_{rst}) must be of Type (II) or (III). Hence, there are distinct children $v_1, v_2 \in \mathsf{child}(\rho)$ with $\sigma(L(T_{rst}(v_1))) = \{r,s\}$ and $\sigma(L(T_{rst}(v_2))) = \{r,t\}$ for distinct colors r,s,t (up to permutation of the colors). Then there exist leaves

$x, y \in L(T_{rst}(v_1))$ and $x', z \in L(T_{rst}(v_2))$ with $x, x' \in L[r]$, $y \in L[s]$, and $z \in L[t]$ such that $xy, x'z \in E(G_{rst})$ (cf. Lemma 5.9). Moreover, Lemma 5.18(ii) implies that $yz \in E(G_{rst})$ as well as $xz, x'y \notin E(G_{rst})$. Hence, $\langle xyzx' \rangle$ is an induced P_4 in (G_{rst}, σ_{rst}) and thus, by Obs. 5.6, also in (G, σ). Since $t \notin \sigma(L(T(v_1)))$, we have $\rho = \mathrm{lca}_T(x, z) \preceq_T \mathrm{lca}_T(x, z')$ for any $z' \in L[t]$, i.e., $z \in N_t^+(x)$ in (T, σ). In particular, $xz \notin E(G)$ then immediately implies $x \notin N_r^+(z)$. One similarly argues that $y \in N_s^+(x')$ and $x' \notin N_r^+(y)$ in (T, σ), hence $(x, z), (x', y) \in E(\vec{G})$ and $(z, x), (y, x') \notin E(\vec{G})$. Therefore $\langle xyzx' \rangle$ is a good quartet in $(\vec{G}, \sigma)$.

Now suppose that (G_{rst}, σ_{rst}) is not S-thin. In this case, we apply the same arguments as above to the quotient graph $(G_{rst}/\mathsf{S}, \sigma_{rst}/\mathsf{S})$ and conclude that there exists a good quartet $\langle [x][y][z][x'] \rangle$ induced by the tree (T_{rst}, σ_{rst}) that explains $(G_{rst}/\mathsf{S}, \sigma_{rst}/\mathsf{S})$. Let $x \in [x], y \in [y], z \in [z]$ and $x' \in [x']$. Lemma 5.4 implies that $\langle xyzx' \rangle$ is an induced P_4 with $\sigma(x) = \sigma(x')$ in (G_{rst}, σ_{rst}) and thus, by Obs. 5.6, also in (G, σ). To conclude that $\langle xyzx' \rangle$ is a good quartet, it remains to show that $(x, z), (x', y) \in E(\vec{G})$ and $(z, x), (y, x') \notin E(\vec{G})$. In order to see this, observe first that $([x], [z]), ([x'], [y]) \in E(\vec{G}(T_{rst}, \sigma_{rst}))$ and $([z], [x]), ([y], [x']) \notin E(\vec{G}(T_{rst}, \sigma_{rst}))$ since $\langle [x][y][z][x'] \rangle$ is a good quartet induced by (T_{rst}, σ_{rst}). To obtain a tree $(\hat{T}, \hat{\sigma})$ that explains (G_{rst}, σ_{rst}), we can proceed as in the proof of the "if-direction" in Lemma 5.6 and simply replace all edges $\mathsf{par}([v])[v]$ in T_{rst} by edges $\mathsf{par}([v])v'$ for all $v' \in [v]$ and putting $\hat{\sigma}(v') = \sigma_{rst}([v])$. Clearly, the latter construction and $([x], [z]), ([x'], [y]) \in E(\vec{G}(T_{rst}, \sigma_{rst}))$ and $([z], [x]), ([y], [x']) \notin E(\vec{G}(T_{rst}, \sigma_{rst}))$ implies that $(x, z), (x', y) \in E(\vec{G}(\hat{T}, \hat{\sigma}))$ and $(z, x), (y, x') \notin E(\vec{G}(\hat{T}, \hat{\sigma}))$. Therefore $\langle xyzx' \rangle$ is a good quartet induced by $(\hat{T}, \hat{\sigma})$ and thus, in (G_{rst}, σ_{rst}). Now Obs. 5.6 implies that $\langle xyzx' \rangle$ is a good quartet in $(\vec{G}, \sigma)$.

□

Since every bad quartet induces in particular an induced P_4 with endpoints of the same color, Lemma 5.37 immediately implies:

Corollary 5.12. *If a BMG $(\vec{G}, \sigma)$ contains a bad quartet, then it contains a good quartet. In particular, any BMG $(\vec{G}, \sigma)$ whose symmetric part contains a 3-RBMG of Type* (B) *or* (C) *as induced subgraph, contains a good quartet.*

Proof. The first statement is an immediate consequence of Lemma 5.37. If (G, σ) is a 3-RBMG of Type (B), then, by definition, it contains an induced P_4 of the form (r, s, t, r) for distinct colors r, s, t. Hence, by Lemma 5.31, this P_4 is either a good or a bad quartet in $(\vec{G}, \sigma)$, where $(\vec{G}, \sigma)$ is a BMG whose symmetric part contains (G, σ) as induced subgraph. Lemma 5.37 thus implies that $(\vec{G}, \sigma)$ contains a good quartet. If (G, σ) is a 3-RBMG of Type (C), it must, by definition, contain an induced C_6 of the form (r, s, t, r, s, t). In particular, it contains an induced P_4 whose endpoints are of the same color, thus we can again apply the same argumentation to complete the proof. □

The converse of Cor. 5.12 is, however, not true. As an example, consider the tree $T = ((x, y), (x'z))$ in Newick format with $\sigma(x) = \sigma(x') = r$, $\sigma(y) = s$, $\sigma(z) = t$. The graph $G(T, \sigma)$ is the P_4 $\langle xyzx' \rangle$, which is of course uniquely

defined. The BMG $\vec{G}(T,\sigma)$ contains the directed edges $(x,z),(x',y)$ but not $(z,x),(y,x')$, hence $Q=\{x,y,z,x'\}=V(T)$ is a good quartet.
We close this section with a variation of Lemma 5.35 for Type (C) RBMGs:

Lemma 5.38. *Let (G,σ) be a connected, S-thin 3-RBMG of Type* (C) *that contains an induced hexagon $H := \langle x_1y_1z_1x_2y_2z_2\rangle$ with $|N_t(x_1)|>1$, where $x_i\in L[r], y_i\in L[s], z_i\in L[t]$ for distinct colors r,s,t. Moreover, let (T,σ) explain (G,σ) and $v := \mathrm{lca}_T(x_1,x_2,y_1,y_2,z_1,z_2)$. Then*

(i) $\langle x_1y_1z_1x_2\rangle$, $\langle z_1x_2y_2z_2\rangle$, and $\langle y_2z_2x_1y_1\rangle$ are good quartets in $\vec{G}(T,\sigma)$,

(ii) $\langle y_1z_1x_2y_2\rangle$, $\langle x_2y_2z_2x_1\rangle$, and $\langle z_2x_1y_1z_1\rangle$ are bad quartets in $\vec{G}(T,\sigma)$, and

(iii) $x_1,y_1\preceq_T v_1$, $x_2,z_1\preceq_T v_2$, and $y_2,z_2\preceq_T v_3$ for some distinct $v_1,v_2,v_3\in \mathsf{child}_T(v)$.

Proof. We start with proving Properties (i) and (ii). By definition and Lemma 5.31, $\langle x_1y_1z_1x_2\rangle$ is either a good or a bad quartet in $\vec{G}(T,\sigma)$. Assume, for contradiction, that it is a bad quartet in $\vec{G}(T,\sigma)$, thus, in particular, $(z_1,x_1)\in E(\vec{G}(T,\sigma))$ and $(x_1,y_1)\notin E(\vec{G}(T,\sigma))$. Hence, as also $\langle z_2x_1y_1z_1\rangle$ must be either a good or a bad quartet in $\vec{G}(T,\sigma)$, this immediately implies that $\langle z_2x_1y_1z_1\rangle$ is a good quartet in $\vec{G}(T,\sigma)$. Let $w := \mathrm{lca}_T(z_2,x_1,y_1,z_1)$. Lemma 5.35 then implies that there exist distinct $w_1,w_2\in\mathsf{child}_T(w)$ such that $z_2,x_1\preceq_T w_1$ and $y_1,z_1\preceq_T w_2$. Clearly, as $x_1y_1\in E(G)$ and $\mathrm{lca}_T(x_1,y_1)=w$, we must have $s\notin\sigma(L(T(w_1)))$. Since $|N_t(x_1)|>1$, there is a leaf $z\in L[t]\setminus\{z_1,z_2\}$ such that $x_1z\in E(G)$. By Lemma 5.9, there exists an edge $x'z'\in E(G(T,\sigma))$ with $x'\in L[r]\cap L(T(w'))$, $z'\in L[t]\cap L(T(w'))$ for any inner vertex $w'\preceq_T w$. One easily verifies that this and $x_1z_2\in E(G)$ necessarily implies that the leaves x_1, z_2, and z must all be incident to the same parent in T. However, we then have $N(z)=N(z_2)$, i.e., z and z_2 belong to the same S-class; a contradiction since (G,σ) is S-thin. We therefore conclude that $\langle x_1y_1z_1x_2\rangle$ must be a good quartet. Hence, $(x_2,y_1),(x_1,z_1)\in E(\vec{G}(T,\sigma))$ and $(y_1,x_2),(z_1,x_1)\notin E(\vec{G}(T,\sigma))$, which, as a consequence of Lemma 5.31, immediately implies that $\langle y_1z_1x_2y_2\rangle$ and $\langle z_2x_1y_1z_1\rangle$ are bad quartets in $\vec{G}(T,\sigma)$. This similarly implies that $\langle z_1x_2y_2z_2\rangle$ and $\langle y_2z_2x_1y_1\rangle$ are good quartets, from which we finally conclude that $\langle x_2y_2z_2x_1\rangle$ is a bad quartet in $\vec{G}(T,\sigma)$.

We continue with showing Property (iii). Property (i) implies that $\langle x_1y_1z_1x_2\rangle$ and $\langle z_1x_2y_2z_2\rangle$ are good quartets in $\vec{G}(T,\sigma)$. Hence, by Lemma 5.35, we have $x_1,y_1\preceq_T w_1$, $x_2,z_1\preceq_T w_2$ for distinct $w_1,w_2\in\mathsf{child}_T(u_1)$, where $u_1 := \mathrm{lca}_T(x_1,y_1,z_1,x_2)$, and $y_2,z_2\preceq_T w_1'$, $x_2,z_1\preceq_T w_2'$ for distinct $w_1',w_2'\in\mathsf{child}_T(u_2)$, where $u_2 := \mathrm{lca}_T(y_2,z_2,z_1,x_2)$, respectively. Since u_1 and u_2 are both located on the path from z_1 to the root of T, they must be comparable. Next, we show that $u_1=u_2$. Assume first, for contradiction, $u_1\prec_T u_2$. Then, by construction, we have $\mathrm{lca}_T(x_2,y_1)=u_1\prec_T u_2=\mathrm{lca}_T(x_2,y_2)$; a contradiction to $x_2y_2\in E(G)$. Similarly, $u_2\prec_T u_1$ yields a contradiction and thus, $u_1=u_2=v$ and, in particular, $w_2=w_2'$. It remains to show $w_1\neq w_1'$. Assume, for contradiction, $w_1=w_1'$. Then $\mathrm{lca}_T(x_1,y_2)\preceq_T w_1\prec_T v=\mathrm{lca}_T(x_2,y_2)$; a contradiction to $x_2y_2\in E(G)$. Hence, w_1, w_2, and w_1' are distinct children of v in T, which completes the proof. □

This section finally combines the results about 3-RBMGs to obtain a characterization of RBMG with an arbitrary number of colors (Subsection 5.7.1). In addition, it gives a characterization of RBMGs that are also cographs (Subsection 5.7.2), which is of particular interest in the context of orthology. It turns out that those are actually equivalent to so-called hc-*cographs* (Subsection 5.7.3), a class of colored graphs that is closely related to cographs. We finally close this section with some algorithmic considerations about the recognition of *hc*-cographs in Subsection 5.7.4.

5.7.1 *The General Case: Combination of 3-RBMGs*

The key idea of characterizing n-RBMGs is to combine the information contained in their 3-colored induced subgraphs (G_{rst}, σ_{rst}). Observation 5.6 plays a major role in this context; it shows that (G_{rst}, σ_{rst}) is an induced subgraph of an n-RBMG and it is always a 3-RBMG that is explained by (T_{rst}, σ_{rst}). Unfortunately, the converse of Observation 5.6 is in general not true. Fig. 28 shows a 4-colored graph that is not a 4-RBMG while each of the four subgraphs induced by a triplet of colors is a 3-RBMG. Observation 5.6 can, however, be rephrased in the following way:

Observation 5.7. *Let (G, σ) be an n-RBMG for some $n \geq 3$. Then (T, σ) explains (G, σ) if and only if (T_{rst}, σ_{rst}) explains (G_{rst}, σ_{rst}) for all triplets of colors $r, s, t \in S$.*

Lemma 5.39. *Let (T_e, σ) be the tree obtained by contracting an inner edge of (T, σ). Then $G(T, \sigma)$ is a subgraph of $G(T_e, \sigma)$.*

Proof. Consider the edge $e = uv$ in T. By construction $(T_e, \sigma) \leq (T, \sigma)$, thus Lemma 4.14 implies $N^+_T(x) \subseteq N^+_{T_e}(x)$ in the BMG $\vec{G}(T, \sigma)$ for all $x \in L(T) \setminus L(T(v))$ and $N^+_T(y) = N^+_{T_e}(y)$ for all $y \in L(T(v))$. It immediately follows $N^-_T(z) \subseteq N^-_{T_e}(z)$ for all $z \in L(T)$. Hence, $E(G) \subseteq E(G(T_e))$. Since the leaf set remains unchanged, $L(T) = L(T_e)$, we conclude that the $G(T, \sigma)$ is a subgraph of $G(T_e, \sigma)$. □

Definition 5.15. *Let (G, σ) be an n-RBMG. Then the* tree set of (G, σ) *is the set $\mathcal{T}(G, \sigma) := \{(T, \sigma) \mid (T, \sigma)$ is least resolved w.r.t. $G(T, \sigma)$ and $G(T, \sigma) = (G, \sigma)\}$ of all leaf-colored trees explaining (G, σ). Furthermore, we write $\mathcal{T}_{rst}(G, \sigma)$ for the set of all least resolved trees explaining the induced subgraphs (G_{rst}, σ_{rst}).*

It is tempting to conjecture that the existence of a supertree for the tree set $\mathfrak{T} := \{T \in \mathcal{T}_{rst}(G_{rst}, \sigma_{rst}), r, s, t \in S\}$ is sufficient for (G, σ) to be an n-RBMG. However, this is not the case as shown by the counterexample in Fig. 28.

Theorem 5.7. *A (not necessarily connected) undirected colored graph (G, σ) is an n-RBMG with $n \geq 3$ if and only if (i) all induced subgraphs (G_{rst}, σ_{rst}) on three colors are 3-RBMGs and (ii) there exists a supertree (T, σ) of the tree set $\mathfrak{T} := \{T \in \mathcal{T}_{rst}(G, \sigma) \mid r, s, t \in S\}$, such that $G(T, \sigma) = (G, \sigma)$.*

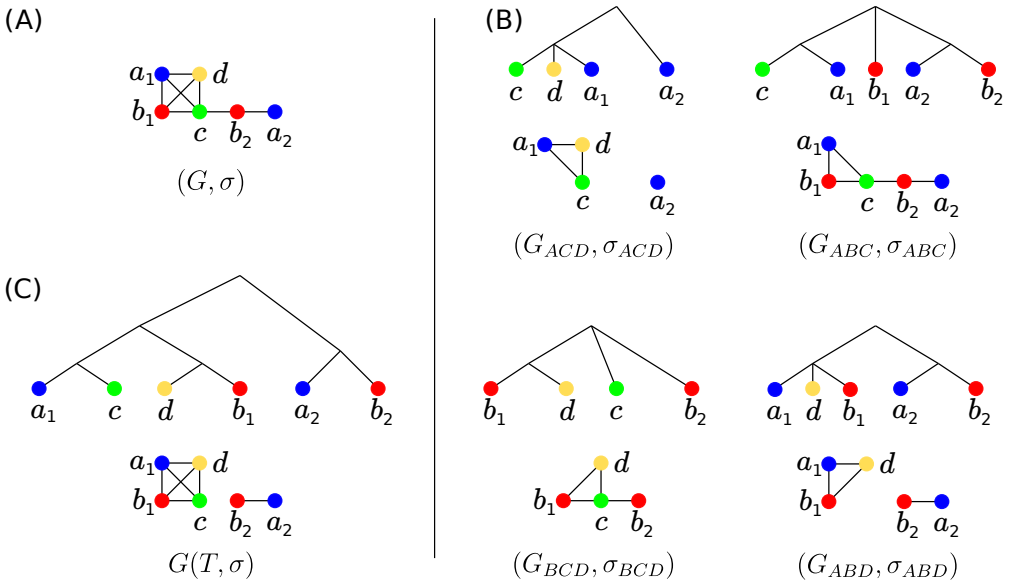


Fig. 28. The 4-colored graph (G, σ) in (A) with color set $S = \{A, B, C, D\}$ is not an RBMG. All four subgraphs induced by three of the four colors, however, are (not necessarily connected) 3-RBMGs. These are explained by the unique least resolved trees in (B). Because of the uniqueness of the least resolved trees on three colors, the tree explaining (G, σ) must display these four trees. The tree (T, σ) in Panel (C) is the least resolved supertree of $\mathfrak{T} := \bigcup_{r,s,t \in S} T_{rst}$. However, (T, σ) does not explain (G, σ) since the edge b_2c is not contained in $G(T, \sigma)$. Clearly, there exists no refinement (T', σ) of (T, σ) such that $b_2c \in E(G(T', \sigma))$ and therefore (G, σ) is not an RBMG.

Proof. Suppose (G, σ) is an n-RBMG, $n \geq 3$, explained by some tree (T, σ). Then Obs. 5.7 implies that (G_{rst}, σ_{rst}) is a 3-RBMG that is explained by (T_{rst}, σ_{rst}) for all triplets of colors $r, s, t \in S$. By definition, each (T_{rst}, σ_{rst}) is displayed by (T, σ) and thus, (T, σ) is a supertree of these trees. Hence, the conditions are necessary. Conversely, the existence of some supertree (T, σ) with $G(T, \sigma) = (G, \sigma)$, i.e., (T, σ) explains (G, σ), clearly implies that (G, σ) is an n-RBMG. □

Whether the recognition problem of n-RBMGs is NP-hard or not may strongly depend on the number of least resolved trees for a given 3-colored induced subgraph. However, even if this number is polynomial bounded in the input size (e.g. number of vertices), the number of possible (least resolved) trees that explain a given n-RBMG, may grow exponentially. In particular, since the order of the inner nodes in the 2-colored subtrees of Type (I), (II), and (III) trees is in general arbitrary, determining the number of least resolved trees seems to be far from trivial. It is therefore left as an open problem at this point.

5.7.2 *Characterization of n-RBMGs that are cographs*

Probably the most important application of reciprocal best matches is orthology detection. Since orthology relations are cographs, it is of particular interest to

characterize RBMGs of this type. Since cographs are hereditary (see e.g. [215] where they are called Hereditary Dacey graphs), one expects their 3-colored restrictions to be of Type (A). The next theorem shows that this intuition is essentially correct. It is based on the following observation about cographs:

Observation 5.8. *Any undirected colored graph (G, σ) is a cograph if and only if the corresponding* S*-thin graph $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$ is a cograph.*

Proof. It directly follows from Lemma 5.4 that (G, σ) contains an induced P_4 if and only if $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$ contains an induced P_4, which yields the statement. □

Note in passing that point-determining (thin) cographs recently have attracted some attention in the literature [148].

Theorem 5.8. *Let (G, σ) be an n-RBMG with $n \geq 3$, and denote by $(G'_{rst}, \sigma'_{rst}) := (G_{rst}/S, \sigma_{rst}/S)$ the* S*-thin version of the 3-RBMG that is obtained by restricting (G, σ) to the colors r, s, and t. Then (G, σ) is a cograph if and only if every* 3-colored *connected component of $(G'_{rst}, \sigma'_{rst})$ is a 3-RBMG of Type* (A) *for all triples of distinct colors r, s, t.*

Proof. We first emphasize that distinct S-classes of some S-thin n-RBMG (G, σ) may belong to the same S-class in $(G'_{rst}, \sigma'_{rst}) := (G_{rst}/S, \sigma_{rst}/S)$. Likewise, distinct S-classes in $(G'_{rst}, \sigma'_{rst})$ may belong to the same S-classes in $(G'_{r's't'}, \sigma'_{r's't'})$. In the following, the vertex set of a connected component $(G^*_{rst}, \sigma^*_{rst})$ of $(G'_{rst}, \sigma'_{rst})$ will be denoted by L^*_{rst}.

Recall that (G, σ) is a cograph if and only if all of its connected components are cographs. Clearly, if (G, σ) is an RBMG, then (G_{rst}, σ_{rst}) and thus, in particular, $(G'_{rst}, \sigma'_{rst})$ is a 3-RBMG (cf. Obs. 5.6) for any three distinct colors r, s, t. Moreover, since (G, σ) is a cograph, it cannot contain an induced P_4, thus its induced subgraph on $L[r] \cup L[s] \cup L[t]$ and therefore also $(G'_{rst}, \sigma'_{rst})$ do not contain an induced P_4 either, i.e., each connected component of $(G'_{rst}, \sigma'_{rst})$ is again a cograph. Hence, by Obs. 5.3, each of the connected components with three colors must be of Type (A).

Conversely, suppose that, for any distinct colors r, s, t, each connected component $(G^*_{rst}, \sigma^*_{rst})$ of $(G'_{rst}, \sigma'_{rst})$ is a 3-RBMG of Type (A). Thus $(G^*_{rst}, \sigma^*_{rst})$ is again S-thin. Obs. 5.3 implies that $(G^*_{rst}, \sigma^*_{rst})$ must be a cograph. Hence, in particular, (G, σ) cannot contain an induced P_4 on two or three colors (cf. Obs. 5.6). Assume, for contradiction, that (G, σ) contains an induced P_4 $\langle abcd \rangle$ on four distinct colors, where $\sigma(a) = A$, $\sigma(b) = B$, $\sigma(c) = C$, and $\sigma(d) = D$. By abuse of notation, we will write a, b, c, and d for the S-classes $[a]$, $[b]$, $[c]$, and $[d]$, respectively. Hence, Lemma 5.4 implies that $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$ contains the induced P_4 $\langle abcd \rangle$ on four distinct colors. By Obs. 5.6, $\langle abc \rangle$ must again be an induced P_3 in some connected component $(G^*_{ABC}, \sigma^*_{ABC})$ of $(G'_{ABC}, \sigma'_{ABC})$. Let L^*_{ABC} be the leaf set of $(G^*_{ABC}, \sigma^*_{ABC})$. Since $G^*_{ABC} \notin \mathcal{P}_3$ as a consequence of Lemma 5.21, $(G^*_{ABC}, \sigma^*_{ABC})$ must contain at least four vertices, i.e., $|L^*_{ABC}| > 3$. Hence, as $(G^*_{ABC}, \sigma^*_{ABC})$ is of Type (A), Lemma 5.21 implies that $(G^*_{ABC}, \sigma^*_{ABC})$ contains a hub-vertex. By Property (A1), the hub-vertex must be connected to any other vertex in $(G^*_{ABC}, \sigma^*_{ABC})$. Hence, neither a nor c

can be the hub-vertex, since $ac \notin E(G)$. By Cor. 5.7, the hub-vertex must be the only vertex of its color in $(G^*_{ABC}, \sigma^*_{ABC})$. Therefore no vertex of color A or C in $(G^*_{ABC}, \sigma^*_{ABC})$ can be the hub-vertex. We therefore conclude that the hub-vertex must be b.

Applying the same argumentation to the connected component $(G^*_{BCD}, \sigma^*_{BCD})$ of $(G'_{BCD}, \sigma'_{BCD})$ that contains the induced P_3 $\langle bcd \rangle$, we can conclude that $|L^*_{BCD}| > 3$ and c must be the hub-vertex of $(G^*_{BCD}, \sigma^*_{BCD})$. Thus, in particular, it is the only vertex of color C in $(G^*_{BCD}, \sigma^*_{BCD})$.

Moreover, since $|L^*_{ABC}| > 3$ and b is the only vertex of color B, there must be at least one other vertex of color A or C in the connected component $(G^*_{ABC}, \sigma^*_{ABC})$.

Assume, for contradiction, that L^*_{ABC} contains a leaf c' of color C such that $c' \neq c$ in $(G^*_{ABC}, \sigma^*_{ABC})$. Since b is the hub-vertex of $(G^*_{ABC}, \sigma^*_{ABC})$, we have $bc' \in E(G^*_{ABC})$ and thus, $bc' \in E(G)$. As c is the only leaf of color C in $(G^*_{BCD}, \sigma^*_{BCD})$, we must ensure $c = c'$ in G^*_{BCD}. Thus $cd \in E(G)$ implies $c'd \in E(G)$. Since $c \neq c'$ in G^*_{ABC}, c must be adjacent to a vertex $\tilde{a}$ of color A that is not adjacent to c', or *vice versa*. Suppose first that $\tilde{a}c' \in E(G)$ and $\tilde{a}c \notin E(G)$. In this case $a = \tilde{a}$ in G^*_{ABC} is possible. Since b is the hub-vertex of $(G^*_{ABC}, \sigma^*_{ABC})$, we have $\tilde{a}b \in E(G)$. Consider $(G'_{ACD}, \sigma'_{ACD})$. By construction, the vertices $\tilde{a}, c, c', d$ are contained in the same connected component $(G^*_{ACD}, \sigma^*_{ACD})$ of Type (A) in the 3-RBMG $(G'_{ACD}, \sigma'_{ACD})$. Since $\tilde{a}c \notin E(G)$, the hub-vertex of $(G^*_{ACD}, \sigma^*_{ACD})$ must be of color D. Hence, as the hub-vertex is the only vertex of its color in $(G^*_{ACD}, \sigma^*_{ACD})$, we can conclude that d is the hub-vertex. Therefore $\tilde{a}d \in E(G)$, which in particular implies $\tilde{a} \neq a$ in $(G^*_{ACD}, \sigma^*_{ACD})$ because $ad \notin E(G)$ by assumption. Hence, $\langle ab\tilde{a}d \rangle \in \mathcal{P}_4$ in $(G^*_{ABD}, \sigma^*_{ABD})$; a contradiction. Now assume $\tilde{a}c \in E(G)$ and $\tilde{a}c' \notin E(G)$. Analogous argumentation implies $\langle ab\tilde{a}d \rangle \in \mathcal{P}_4$ in $(G^*_{ABD}, \sigma^*_{ABD})$; again a contradiction.

We therefore conclude that L^*_{ABC} does not contain a leaf $c' \neq c$ of color C and hence, $L^*_{ABC}[C] = \{c\}$. Together with $L^*_{ABC}[B] = \{b\}$ and $|L^*_{ABC}| > 3$, this implies that there must exist a leaf $a' \neq a$ of color A in L^*_{ABC}. We have $a'b \in E(G)$ because b is the hub-vertex of $(G^*_{ABC}, \sigma^*_{ABC})$. Hence, since $(G^*_{ABC}, \sigma^*_{ABC})$ is S-thin, the neighborhoods of a and a' must differ. The latter and $L^*_{ABC}[B] \cup L^*_{ABC}[C] = \{b, c\}$ implies $a'c \in E(G^*_{ABC})$, i.e., $a'c \in E(G)$. One now easily checks that, by S-thinness of $(G^*_{ABC}, \sigma^*_{ABC})$, there cannot be a third vertex of color A in L^*_{ABC}, thus $L^*_{ABC} = \{a, a', b, c\}$. Applying analogous arguments to $(G^*_{BCD}, \sigma^*_{BCD})$ shows $L^*_{BCD} = \{b, c, d, d'\}$, where $\sigma(d') = D$ and $bd', cd' \in E(G)$. Moreover, we have $a'd \notin E(G)$, because otherwise $ad, bd \notin E(G)$ would imply that $\langle aba'd \rangle$ is an induced P_4 in $(G^*_{ABD}, \sigma^*_{ABD})$; a contradiction since $(G^*_{ABD}, \sigma^*_{ABD})$ is of Type (A). Similarly, $ad' \notin E(G)$ as otherwise $(G^*_{ACD}, \sigma^*_{ACD})$ would contain the induced P_4 $\langle ad'cd \rangle$.

In summary, $\langle abcd \rangle$ is an induced P_4 in (G, σ) and there are vertices a' and d' such that $a'b, a'c, d'b, d'c \in E(G)$ and $a'd, ad' \notin E(G)$, see Fig. 29(A).

Let (T, σ) be a tree that explains (G, σ) and $u := \mathrm{lca}(b, c)$. The steps of the subsequent proof are illustrated in Fig. 29. Since $ab \in E(G)$, we have $\mathrm{lca}(a, b) \preceq_T \mathrm{lca}(a', b)$. Similarly, $a'b \in E(G)$ implies $\mathrm{lca}(a', b) \preceq_T \mathrm{lca}(a, b)$. Hence, we clearly have $v := \mathrm{lca}(a, b) = \mathrm{lca}(a', b)$. Note that v and u are both

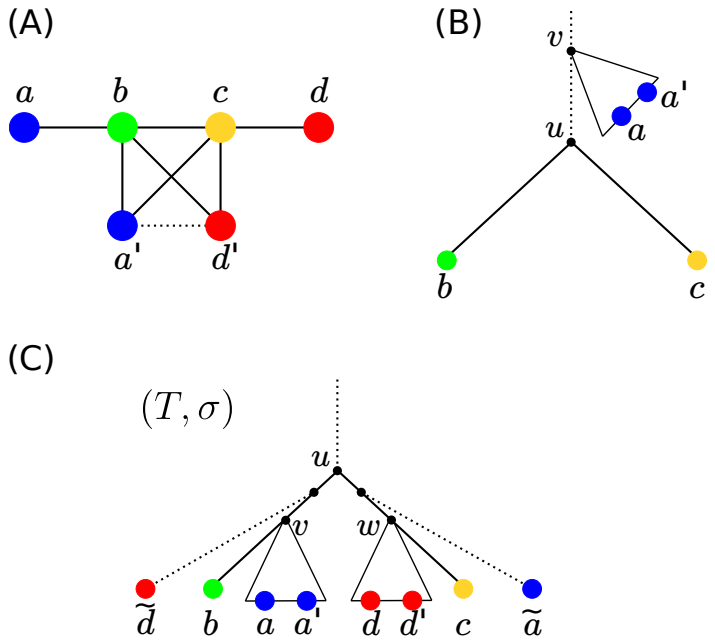


Fig. 29. Panel (A) shows the induced subgraph $(G_{|L'}, \sigma_{|L'})$ with $L' = \{a, a', b, c, d, d'\}$ of (G, σ) that is used in the proof of Thm. 5.8. In the two trees in Panels (B) and (C), it holds $u := \mathrm{lca}(b, c)$ and $v := \mathrm{lca}(a, b) = \mathrm{lca}(a', b)$. Panel (B) shows a sketch of the subtree of (T, σ) in case $v \succeq_T u$. Panel (C) shows a sketch of a possible subtree of (T, σ) in case $v \prec_T u$ and $w := \mathrm{lca}(c, d) = \mathrm{lca}(c, d') \prec_T u$. In this representation of (T, σ) we have $\mathrm{lca}(a, \widetilde{d}) \succ_T v$ and $\mathrm{lca}(\widetilde{a}, d) \succ_T w$. However, $\mathrm{lca}(a, \widetilde{d}) \preceq_T v$ or $\mathrm{lca}(\widetilde{a}, d) \preceq_T w$ may be possible. Dashed lines represent edges in $(G_{|L'}, \sigma_{|L'})$ and paths in the trees that may or may not be present. Solid lines in the trees represent paths.

ancestors of b and are thus located on the path from the root ρ_T to b. In other words, v and u are always comparable in T, i.e., we have either $v \succeq_T u$ or $v \prec_T u$. If $v \succeq_T u$, then $v = \mathrm{lca}(a, c) = \mathrm{lca}(a', c)$. Since $a'c \in E(G)$, we have $v \preceq_T \mathrm{lca}(a', \widetilde{c})$ and $v \preceq_T \mathrm{lca}(\widetilde{a}, c)$ for all $\widetilde{a} \in L[A]$ and all $\widetilde{c} \in L[C]$. Together with $v = \mathrm{lca}(a, c)$, this implies $ac \in E(G)$; a contradiction. Thus only the case $v \prec_T u$ is possible and hence, v must be located on the path from some child of u to b in T. Similarly, we have $w := \mathrm{lca}(c, d) = \mathrm{lca}(c, d')$ because $cd, cd' \in E(G)$. As $bd' \in E(G)$, $bd \notin E(G)$, we can apply an analogous argumentation as for u and v to conclude that only the case $w \prec_T u$ is possible. Thus w must be located on the path from some child of u to c in T. In particular, we have $\mathrm{lca}(v, w) = u$ by definition of u and therefore, $u = \mathrm{lca}(a, d)$. Since $ad \notin E(G)$, we have $u = \mathrm{lca}(a, d) \succ_T \mathrm{lca}(a, \widetilde{d})$ for some $\widetilde{a} \in L[A]$ or $u = \mathrm{lca}(a, d) \succ_T \mathrm{lca}(\widetilde{a}, d)$ for some $\widetilde{d} \in L[D]$. Assume $u = \mathrm{lca}(a, d) \succ_T \mathrm{lca}(a, \widetilde{d})$ for some $\widetilde{d} \in L[D]$. In this case, $\mathrm{lca}(a, \widetilde{d})$ must be located on the path from some child u' of u to a. Thus $u \succ_T u' \succ_T a, \widetilde{d}$ and by construction, $u' \succ_T b$. Hence, $\mathrm{lca}(b, \widetilde{d}) \prec_T u = \mathrm{lca}(b, d')$ and thus, $bd' \notin E(G)$; a contradiction. Analogously, $u = \mathrm{lca}(a, d) \succ_T \mathrm{lca}(\widetilde{a}, d)$ would imply that $\mathrm{lca}(\widetilde{a}, d)$ is located on the path from some child of u to d, which would contradict $a'c \in E(G)$. Therefore the tree (T, σ) does not explain (G, σ); a contradiction.

Thus, if for any distinct colors r, s, t, each 3-colored connected component $(G^*_{rst}, \sigma^*_{rst})$ of $(G'_{rst}, \sigma'_{rst})$ is a 3-RBMG of Type (A), then (G, σ) must be a cograph. □

For technical reasons, the latter result has been stated for S-thin induced 3-RBMGs only. However, due to Lemma 5.4, it clearly extends to general RBMGs:

Corollary 5.13. *An n-RBMG (G, σ) with $n \geq 3$ is a cograph if and only if every 3-colored connected component of (G_{rst}, σ_{rst}) is a Type* (A) *3-RBMG for all triplets of distinct colors r, s, t.*

Moreover, as a by-product of the previous proof, the induced subgraph shown in Panel (A) of Fig. 29 is a forbidden subgraph of general RBMGs.

5.7.3 *Hierarchically Colored Cographs*

Thm. 5.8 yields a polynomial time algorithm for recognizing n-RBMGs that are cographs. It is not helpful, however, for the reconstruction of a tree (T, σ) that explains such a graph. In this part, we derive an alternative characterization in terms of so-called hierarchically colored cographs (*hc*-cographs). As we shall see, the cotrees of *hc*-cographs explain a given n-RBMG and can be constructed in polynomial time.

Definition 5.16. *A graph that is both a cograph and an RBMG is a* co-RBMG.

Interestingly, for every leaf-labeled tree explaining a co-RBMG, one can identify certain subtrees whose color sets do not overlap. This property forms the basis for the definition of so-called hc-*cographs*.

Lemma 5.40. *Let (G, σ) be a co-RBMG that is a explained by (T, σ). Then, for any $v \in V^0(T)$ and each pair of distinct children $w_1, w_2 \in \mathsf{child}_T(v)$, the sets $\sigma(L(T(w_1)))$ and $\sigma(L(T(w_2)))$ do not overlap.*

Proof. Assume, for contradiction, that there exists some $v \in V^0(T)$ and distinct $w_1, w_2 \in \mathsf{child}(v)$ such that $r \in \sigma(L(T(w_1))) \cap \sigma(L(T(w_2)))$, s is contained in $\sigma(L(T(w_1)))$ but not in $\sigma(L(T(w_2)))$, and t is contained in $\sigma(L(T(w_2)))$ but not in $\sigma(L(T(w_1)))$ for three distinct colors r, s, t in (G, σ). Then, by Cor. 5.3, there is a pair $x, y \in L(T(w_1))$ with $\sigma(x) = r$, $\sigma(y) = s$ such that $xy \in E(G)$, as well as a pair $x', z \in L(T(w_2))$ with $\sigma(x') = r, \sigma(z) = t$ such that $x'z \in E(G)$. Since $t \notin \sigma(L(T(w_1)))$ and $s \notin \sigma(L(T(w_2)))$, we have $\mathrm{lca}_T(y, z) = v \preceq_T \mathrm{lca}_T(y, z')$ for any $z' \in L[t]$ and $\mathrm{lca}_T(y, z) = v \preceq_T \mathrm{lca}_T(y', z)$ for any $y' \in L[s]$, hence $yz \in E(G)$. Moreover, as $\mathrm{lca}_T(z, x') \prec_T v = \mathrm{lca}_T(z, x)$ and $\mathrm{lca}_T(y, x) \prec_T v = \mathrm{lca}_T(y, x')$, the edges xz and $x'y$ are not contained in (G, σ). We therefore conclude that $\langle xyzx' \rangle$ is an induced P_4 in (G, σ); a contradiction since (G, σ) is a cograph.

□

Cographs are constructed using joins and disjoint unions. Motivated by the latter result, we extend these graph operations to vertex-colored graphs:

Definition 5.17. *Let* (H_1, σ_{H_1}) *and* (H_2, σ_{H_2}) *be two vertex-disjoint colored graphs. Then* $(H_1, \sigma_{H_1}) \triangledown (H_2, \sigma_{H_2}) := (H_1 \triangledown H_2, \sigma)$ *and* $(H_1, \sigma_{H_1}) \cup (H_2, \sigma_{H_2}) := (H_1 \cup H_2, \sigma)$ *denotes their join and union, respectively, where* $\sigma(x) = \sigma_{H_i}(x)$ *for every* $x \in V(H_i)$*,* $i \in \{1, 2\}$*.*

It turns out that the join and union of RBMGs (H, σ_H) and $(H', \sigma_{H'})$ with non-overlapping color sets again yield an RBMG (G, σ). Moreover, a tree representation for (G, σ) can be readily obtained by combining the tree representations of (H, σ_H) and $(H', \sigma_{H'})$.

Lemma 5.41. *Let* (G, σ) *be a properly colored, undirected graph such that either* $(G, \sigma) = (H, \sigma_H) \triangledown (H', \sigma_{H'})$ *with* $\sigma(V(H)) \cap \sigma(V(H')) = \emptyset$ *or* $(G, \sigma) = (H, \sigma_H) \cup (H', \sigma_{H'})$ *with* $\sigma(V(H)) \cap \sigma(V(H')) \in \{\sigma(V(H)), \sigma(V(H'))\}$*, where* (H, σ_H) *and* $(H', \sigma_{H'})$ *are disjoint RBMGs. Then* (G, σ) *is an RBMG.*

Moreover, let (H, σ_H) *and* $(H', \sigma_{H'})$ *be explained by the trees* (T_H, σ_H) *and* $(T_{H'}, \sigma_{H'})$*, respectively, and let* (T, σ) *be the tree obtained by joining* (T_H, σ_H) *and* $(T_{H'}, \sigma_{H'})$ *by a common root* ρ_T*. Then* (T, σ) *explains* (G, σ)*.*

Proof. Let (T, σ) be the tree that is obtained by joining (T_H, σ_H) and $(T_{H'}, \sigma_{H'})$ with roots ρ_H and $\rho_{H'}$, respectively, under a common root ρ_T. By construction, $\sigma(V(H)) = \sigma_H$ and $\sigma(V(H')) = \sigma_{H'}$. We first show that $(T(\rho_H), \sigma_H)$ and $(T(\rho_{H'}), \sigma_{H'})$ explain (H, σ_H) and $(H', \sigma_{H'})$, respectively. By construction, we have $\mathrm{lca}_T(x, y) = \mathrm{lca}_{T_H}(x, y)$ and $\mathrm{lca}_T(x, y) \prec_T \mathrm{lca}_T(x, z)$ for any $x, y \in V(H)$ and $z \in V(H')$. It is therefore easy to see that $G(T(\rho_H), \sigma_H) = (H, \sigma_H)$. Analogous arguments show $G(T(\rho_{H'}), \sigma_{H'}) = (H', \sigma_{H'})$. Therefore, in order to show that (T, σ) explains (G, σ), it remains to show that all edges between vertices in $V(H)$ and $V(H')$ are identical in (G, σ) and $G(T, \sigma)$.

Suppose first $(G, \sigma) = (H, \sigma_H) \triangledown (H', \sigma_{H'})$ with $\sigma(V(H)) \cap \sigma(V(H')) = \emptyset$. Thus we need to show $xy \in E(G(T, \sigma))$ for any $x \in L(T(\rho_H))$ and $y \in L(T(\rho_{H'}))$. Since $\sigma(V(H))$ and $\sigma(V(H'))$ form a partition of $\sigma(V(G))$, we have $\mathrm{lca}_T(x, y) = \rho_T$ for any $x \in L[r], y \in L[s]$ with $r \in \sigma(V(H))$ and $s \in \sigma(V(H'))$. Hence, $xy \in E(G(T, \sigma))$ for any $x \in L(T(\rho_H)), y \in L(T(\rho_{H'}))$ and therefore, $G(T, \sigma) = (G, \sigma)$.

Now suppose that $(G, \sigma) = (H, \sigma_H) \cup (H', \sigma_{H'})$ with $\sigma(V(H)) \cap \sigma(V(H')) \in \{\sigma(V(H)), \sigma(V(H'))\}$. Thus we need to show $xy \notin E(G(T, \sigma))$ for any $x \in L(T(\rho_H)), y \in L(T(\rho_{H'}))$. W.l.o.g. assume $\sigma(V(H')) \subseteq \sigma(V(H))$. Hence, $(T(\rho_H), \sigma_H)$ is color-complete. We can therefore apply Lemma 5.12 to conclude that $xy \notin E(G(T, \sigma))$ for any $x \in V(H), y \in V(H')$. Hence, $G(T, \sigma) = (G, \sigma)$. □

The previous result serves as motivation for the definition of *hc*-cographs:

Definition 5.18. *An undirected colored graph* (G, σ) *is a* hierarchically colored cograph (*hc*-cograph) *if*

(K1) $(G, \sigma) = (K_1, \sigma)$*, i.e., a colored vertex, or*

(K2) $(G, \sigma) = (H, \sigma_H) \triangledown (H', \sigma_{H'})$ *and* $\sigma(V(H)) \cap \sigma(V(H')) = \emptyset$*, or*

(K3) $(G, \sigma) = (H, \sigma_H) \cup (H', \sigma_{H'})$ *and* $\sigma(V(H)) \cap \sigma(V(H')) \in \{\sigma(V(H)), \sigma(V(H'))\}$*,*

where both (H, σ_H) *and* $(H', \sigma_{H'})$ *are* hc*-cographs. For the color-constraints (cc) in* (K2) *and* (K3)*, we simply write* (K2cc) *and* (K3cc)*, respectively.*

Omitting the color-constraints reduces Def. 5.18 to Def. 3.1. Therefore we have

Observation 5.9. *If* (G, σ) *is an* hc*-cograph, then* G *is cograph.*

The recursive construction of an *hc*-cograph (G, σ) according to Def. 5.18 immediately produces a binary *hc*-cotree T_{hc}^G corresponding to (G, σ). The construction is essentially the same as for the cotree of a cograph (cf. Corneil et al. [38, Section 3]): Each of its inner vertices is labeled by 1 for a $\triangledown$ operation and 0 for a disjoint union $\cup$, depending on whether (K2) or (K3) is used in the construction step. We write $t : V^0(T_{hc}^G) \to \{0, 1\}$ for the labeling of the inner vertices. The recursion terminates with a leaf of T_{hc}^G whenever a colored single-vertex graph, i.e., (K1) is reached. We therefore identify the leaves of T_{hc}^G with the vertices of (G, σ). The binary *hc*-cograph (T_{hc}^G, t, σ) with leaf coloring σ and labeling t at its inner vertices uniquely determines (G, σ), i.e., $xy \in E(G)$ if and only if $t(\mathrm{lca}(x, y)) = 1$.

By construction, (T_{hc}^G, t) is a not necessarily discriminating cotree for G. An example for different constructions of (T_{hc}^G, t, σ) based on the particular *hc*-cograph representation of (G, σ) is given later in Fig. 30.

While the cograph property is hereditary, this is no longer true for *hc*-cographs, i.e., an *hc*-cograph may contain induced subgraphs that are not *hc*-cographs. As an example, consider the three single vertex graphs (G_i, σ_i) with $V_i = \{i\}$ and colors $\sigma_1(x) = r$ and $\sigma_2(y) = \sigma_3(z) = s \neq r$. Then $(G, \sigma) = ((G_1, \sigma_1) \triangledown (G_2, \sigma_2)) \cup (G_3, \sigma_3)$ is an *hc*-cograph. However, the induced subgraph $(G, \sigma)[x, z] = (G_1, \sigma_1) \cup (G_3, \sigma_3)$ is not an *hc*-cograph, since $\sigma_1(V_1) \cap \sigma_3(V_3) = \emptyset$ and hence, $(G, \sigma)[x, z]$ does not satisfy Property (K3cc).

Both $\triangledown$ and $\cup$ are commutative and associative operations on graphs. For a given cograph G, hence, alternative binary cotrees may exist that can be transformed into each other by applying the commutative or associative laws. This is no longer true for *hc*-cographs as a consequence of the color constraints. There are no restrictions on commutativity, i.e., if (G, σ) can be obtained as the join $(H, \sigma_H) \triangledown (H', \sigma_{H'})$, equivalently we have $(G, \sigma) = (H', \sigma_{H'}) \triangledown (H, \sigma_H)$. The same holds for the disjoint union $\cup$. If (G, σ) is obtained as $(H, \sigma_H) \triangledown \Big((H', \sigma_{H'}) \triangledown (H'', \sigma_{H''})\Big)$, i.e., if $(H', \sigma_{H'}) \triangledown (H'', \sigma_{H''})$ is also an *hc*-cograph, then the color sets of H, H', and H'' must be disjoint by Def. 5.18 and thus, $(H, \sigma_H) \triangledown (H', \sigma_{H'})$ is also an *hc*-cograph. Condition (K3cc), however, is not so well-behaved:

Example: Consider the single vertex graphs (G_i, σ_i) with vertex set $V_i = \{i\}$, $1 \leq i \leq 4$ and colors $\sigma(i) = r$ if i is odd and $\sigma(i) = s \neq r$ if i is even. Consider the graph $G = G_1 \cup (G_2 \cup (G_3 \triangledown G_4))$. By construction, $(G_3 \triangledown G_4, \sigma_{|\{3,4\}})$ is an *hc*-cograph because $\sigma(V_3) \cap \sigma(V_4) = \emptyset$ and thus, (K2cc) is satisfied. Furthermore, $(G_2 \cup (G_3 \triangledown G_4)), \sigma_{|\{2,3,4\}})$ is an *hc*-cograph since $\sigma(V_2) = \{s\} \subseteq \sigma(V_3 \cup V_4) = \{r, s\}$ and thus, (K3cc) is satisfied. Checking (K3cc) again, we verify that (G, σ) is an *hc*-cograph. By associativity of $\triangledown$ and $\cup$, we also have $G' = (G_1 \cup G_2) \cup (G_3 \triangledown G_4) = G$. However, $(G_1 \cup G_2, \sigma_{\{1,2\}})$ is not an *hc*-cograph because $\sigma(V_1) \cap \sigma(V_2) = \emptyset$ implies that $(G_1 \cup G_2, \sigma_{\{1,2\}})$ does not satisfy Property (K3cc).

As a consequence, we cannot simply contract edges in the *hc*-cotree T_{hc}^G with incident vertices labeled by the $\cup$ operation. In other words, it is not sufficient to use discriminating trees to represent *hc*-cotrees. Moreover, not every (binary) tree with colored leaves and internal vertices labeled with ∇ or $\cup$ (which specifies a cograph) determines an *hc*-cotree because in addition the color-restrictions (K2cc) and (K3cc) must be satisfied for each internal vertex.

Lemma 5.42. *Every* hc*-cograph* (G, σ) *is a properly colored cograph.*

Proof. Let (G, σ) be an *hc*-cograph. In order to see that (G, σ) is properly colored, observe that any edge xy in (G, σ) must be the result of some (possible preceding) join $(H, \sigma_H) \nabla (H', \sigma_{H'})$ during the recursive construction of (G, σ) such that $x \in V(H)$ and $y \in V(H')$. Condition (K2) implies that $\sigma(V(H)) \cap \sigma(V(H')) = \emptyset$ and hence, $\sigma(x) \neq \sigma(y)$ for every edge xy in (G, σ). □

Not every properly colored cograph is an *hc*-cograph, however. The simplest counterexample is $\overline{K_2} = K_1 \cup K_1$ with two differently colored vertices, violating (K3cc). The simplest connected counterexample is the 3-colored P_3 since the decomposition $P_3 = (K_1 \cup K_1) \nabla K_1$ is unique, and involves the non-*hc*-cograph $K_1 \cup K_1$ with two distinct colors as a factor in the join.

Theorem 5.9. *A vertex-labeled graph* (G, σ) *is a co-RBMG if and only if it is an* hc*-cograph.*

Proof. Suppose that $(G = (V, E), \sigma)$ with vertex set V and edge set E is a co-RBMG. We show by induction on $|V|$ that (G, σ) is an *hc*-cograph. This is trivially true for the base case $|V| = 1$.

For the induction step assume that any co-RBMG with less than N vertices is at the same time an *hc*-cograph and consider a co-RBMG with $|V| = N$. Since (G, σ) is an n-RBMG, there exists a tree (T, σ) with root ρ_T that explains (G, σ). By Lemma 5.40, none of the color sets $\sigma(L(T(v)))$ and $\sigma(L(T(w)))$ overlap for any two distinct children $v, w \in \mathsf{child}(\rho_T)$. Moreover, Lemma 5.40 allows us to define a partition Π of $\mathsf{child}(\rho_T)$ into classes $P_1, \ldots, P_k$ such that each pair of vertices $v \in P_i$ and $w \in P_j$, $i \neq j$, satisfies $\sigma(L(T(v))) \cap \sigma(L(T(w))) = \emptyset$. Note that each P_i may contain distinct elements v, w such that $\sigma(L(T(v))) \cap \sigma(L(T(w))) \in \{\emptyset, \sigma(L(T(v))), \sigma(L(T(w)))\}$.

First assume that the partition Π of $\mathsf{child}(\rho_T)$ is trivial, i.e., it consists of a single class P_1. Since none of the sets $\sigma(L(T(v)))$ and $\sigma(L(T(w)))$ overlap for any $v, w \in P_1$, there is an element $w \in P_1$ such that $\sigma(L(T(w)))$ is inclusion-maximal, i.e., $\sigma(L(T(v))) \subseteq \sigma(L(T(w)))$ for all $v \in P_1 = \mathsf{child}(\rho_T)$. Let $L_{\neg w} = \bigcup_{v \in P_1 \setminus w} L(T(v))$ and $L_w = L(T(w))$. Since ρ_T is always color-complete, w must be color-complete, i.e., $\sigma(L_w) = \sigma(V)$. Hence, we have $\sigma(L_{\neg w}) \subseteq \sigma(L_w)$.

We continue by showing that $(G[L_{\neg w}], \sigma_{|L_{\neg w}})$ and $(G[L_w], \sigma_{|L_w})$ are RBMGs that are explained by $(T_{|L_{\neg w}}, \sigma_{|L_{\neg w}})$ and $(T_{|L_w}, \sigma_{|L_w})$, respectively. By construction, we have $\mathrm{lca}_T(x, y) = \mathrm{lca}_{T_{|L_{\neg w}}}(x, y)$ and $\mathrm{lca}_T(x, y) \preceq_T \mathrm{lca}_T(x, z)$ for any $x, y \in L_{\neg w}$ and $z \in L_w$. It is therefore easy to see that $G(T_{|L_{\neg w}}, \sigma_{|L_{\neg w}}) = (G[L_{\neg w}], \sigma_{|L_{\neg w}})$. Analogous arguments show that $G(T_{|L_w}, \sigma_{|L_w}) = (G[L_w], \sigma_{|L_w})$. Hence, $(G[L_{\neg w}], \sigma_{|L_{\neg w}})$ and $(G[L_w], \sigma_{|L_w})$ are RBMGs. Since any induced subgraph of a cograph is again a cograph, we can

thus conclude that $(G[L_{\neg w}], \sigma_{|L_{\neg w}})$ and $(G[L_w], \sigma_{|L_w})$ are co-RBMGs. Hence, by induction hypothesis, $(G[L_{\neg w}], \sigma_{|L_{\neg w}})$ and $(G[L_w], \sigma_{|L_w})$ are *hc*-cographs. Moreover, since w is color-complete, we can apply Lemma 5.12 to conclude that $(G, \sigma) = (G[L_{\neg w}], \sigma_{|L_{\neg w}}) \cup (G[L_w], \sigma_{|L_w})$ is the disjoint union of two *hc*-cographs and in addition satisfies $\sigma(L_{\neg w}) \subseteq \sigma(L_w)$. Hence, (G, σ) satisfies Property (K3) and is therefore an *hc*-cograph.

Now assume that Π is non-trivial, i.e., there are at least two classes $P_1, \ldots, P_k$. Then, by construction, we have $\sigma(L(T(v))) \cap \sigma(L(T(w))) = \emptyset$ for all $v \in P_i$ and $w \in P_j$, $i \neq j$. Let $L_i = \bigcup_{v \in P_i} L(T(v))$. By construction, $\sigma(L_i) \cap \sigma(L_j) = \emptyset$ for all distinct i, j. Hence, $\sigma(L_1), \ldots, \sigma(L_k)$ form a partition of $\sigma(V)$. Thus we have $\mathrm{lca}_T(x, y) = \rho_T$ for any $x \in L[r], y \in L[s]$ with $r \in \sigma(L_i)$ and $s \in \sigma(L_j)$ for distinct $i, j \in \{1, \ldots, k\}$, which clearly implies $xy \in E(G)$. Hence, $G = G[L_1] \bigtriangledown G[L_2] \bigtriangledown \ldots \bigtriangledown G[L_k]$. Thus, by setting $H = \bigtriangledown_{i=1}^{k-1} G[L_i]$ and $H' = G[L_k]$, we obtain $(G, \sigma) = (H, \sigma_{|V(H)}) \bigtriangledown (H', \sigma_{|V(H')})$. We proceed to show that $(H, \sigma_{|V(H)})$ and $(H', \sigma_{|V(H')})$ are *hc*-cographs. Since any induced subgraph of a cograph is again a cograph, we can conclude that H and H' are cographs. Thus it remains to show that $(H, \sigma_{|V(H)})$ and $(H', \sigma_{|V(H')})$ are RBMGs. By similar arguments as in the case for one class P_1, one shows that $(T_{|V(H)}, \sigma_{|V(H)})$ and $(T_{|V(H')}, \sigma_{|V(H')})$ explain $(H, \sigma_{|V(H)})$ and $(H', \sigma_{|V(H')})$, respectively. In summary, $(H, \sigma_{|V(H)})$ and $(H', \sigma_{|V(H')})$ are co-RBMGs and thus, by induction hypothesis, *hc*-cographs. Since $\sigma(V(H)) = \bigcup_{i=1}^{k-1} \sigma(L_i)$, $\sigma(V(H')) = \sigma(L_k)$, and $\sigma(L_i), \sigma(L_j)$ are disjoint for distinct $i, j \in \{1, \ldots, k\}$, we can conclude that $\sigma(V(H)) \cap \sigma(V(H')) = \emptyset$. Hence, $(G, \sigma) = (H, \sigma_{|V(H)}) \bigtriangledown (H', \sigma_{|V(H')})$ satisfies Property (K2). Thus (G, σ) is an *hc*-cograph.

Now suppose that $(G = (V, E), \sigma)$ is an *hc*-cograph. Lemma 5.42 implies that G is a cograph. In order to show that (G, σ) is an RBMG, we proceed again by induction on $|V|$. The base case $|V| = 1$ is trivially satisfied. For the induction hypothesis, assume that any *hc*-cograph (G, σ) with $|V| < N$ is an RBMG. Now let (G, σ) with $|V| = N > 1$ be an *hc*-cograph. By definition of *hc*-cographs and since $|V| > 1$, there exist disjoint *hc*-cographs (H, σ_H) and $(H', \sigma_{H'})$ such that either (i) $(G, \sigma) = (H, \sigma_H) \bigtriangledown (H', \sigma_{H'})$ and $\sigma(V(H)) \cap \sigma(V(H')) = \emptyset$, or (ii) $(G, \sigma) = (H, \sigma_H) \cup (H', \sigma_{H'})$ and $\sigma(V(H)) \cap \sigma(V(H')) \in \{\sigma(V(H)), \sigma(V(H'))\}$. By induction hypothesis, $(H, \sigma(V(H)))$ and $(H', \sigma(V(H')))$ are RBMGs. Hence, we can apply Lemma 5.41 to conclude that (G, σ) an RBMG. □

Theorem 5.10. *Every co-RBMG (G, σ) is explained by its cotree $(T_{\mathrm{hc}}^G, \sigma)$.*

Proof. We show by induction on $|V|$ that (G, σ) is explained by (T_{hc}^G, σ). This is trivially true for the base case $|V| = 1$. Assume that any co-RBMG with less than N vertices is explained by its cotree (T_{hc}^G, σ). Now let $(G = (V, E), \sigma)$ be a co-RBMG with $|V| = N$. By Thm. 5.9, (G, σ) is an *hc*-cograph. Thus $(G, \sigma) = (H, \sigma_H) \star (H', \sigma_{H'})$ with $\star \in \{\bigtriangledown, \cup\}$ such that (K2) and (K3), resp., are satisfied. Thm. 5.9 implies that (H, σ_H) and $(H', \sigma_{H'})$ are co-RBMGs. By induction hypothesis, the co-RBMGs (H, σ_H) and $(H', \sigma_{H'})$ are explained by their *hc*-cotrees (T_{hc}^H, σ_H) and $(T_{hc}^{H'}, \sigma_{H'})$, respectively. By construction, (T_{hc}^G, σ) is the tree that is obtained by joining (T_{hc}^H, σ_H) and $(T_{hc}^{H'}, \sigma_{H'})$ under a common root. Lemma 5.41 now implies that the (T_{hc}^G, σ) explains (G, σ).

□

5.7.4 *Recognition of* hc*-cographs*

Although (discriminating) cotrees can be constructed and cographs can be recognized in linear time [86, 24, 37], these results cannot be applied directly to the construction of *hc*-cotrees and the recognition of *hc*-cographs. The key problem is that whenever (G, σ) comprises $k > 2$ connected components, there are $2^{k-1} - 1$ bipartitions $(G, \sigma) = (G_1, \sigma_1) \cup (G_2, \sigma_2)$. For each of them (K3cc) needs to be checked and, if it is satisfied, both (G_1, σ_1) and (G_2, σ_2) need to be tested for being *hc*-cotrees. In general, this incurs exponential effort. The following results show, however, that it suffices to consider a single, carefully chosen bipartition for each disconnected graph (G, σ).

Lemma 5.43. *Every connected component of an* hc*-cograph is an* hc*-cograph.*

Proof. Let (G, σ) be an *hc*-cograph. By Thm. 5.9, (G, σ) is a co-RBMG. Since each connected component of a cograph is again a cograph, each connected component of (G, σ) must be a cograph. In addition, Thm. 5.3 implies that each connected component of (G, σ) is an RBMG. The latter two arguments imply that each connected component of (G, σ) is a co-RBMG. Hence, Thm. 5.9 implies that each connected component of (G, σ) is an *hc*-cograph. □

Lemma 5.44. *Let (G, σ) be a disconnected* hc*-cograph with connected components $G_1 = (V_1, E_1)$, ..., $G_k = (V_k, E_k)$ and let G_ℓ, $1 \leq \ell \leq k$ be a connected component whose color set is minimal w.r.t. inclusion, i.e., there is no $i \in \{1, \dots, k\}$ with $i \neq \ell$ such that $\sigma(V_i) \subsetneq \sigma(V_\ell)$. Denote by $G - G_\ell$ the graph obtained from (G, σ) by deleting the connected component G_ℓ. Then*

$$(G, \sigma) = (G_\ell, \sigma_{|V_\ell}) \cup (G - G_\ell, \sigma_{|V \setminus V_\ell})$$

satisfies Property (K3)*.*

Proof. Let $(G = (V, E), \sigma)$ be a disconnected *hc*-cograph with connected components $(G_1 = (V_1, E_1), \sigma_1), \dots, (G_k = (V_k, E_k), \sigma_k)$ and put $\sigma_i := \sigma_{|V_i}$ for $1 \leq i \leq k$. Let (G_ℓ, σ_ℓ), $1 \leq \ell \leq k$ be a graph such that $\sigma(V_\ell)$ is minimal w.r.t. inclusion. We write $G' = (V', E') := G - G_\ell$ and $\sigma' := \sigma_{|V \setminus V_\ell}$, thus $(G', \sigma') = (G - G_\ell, \sigma_{|V \setminus V_\ell})$ and $V' = V \setminus V_\ell$. In order to prove that $(G, \sigma) = (G_\ell, \sigma_\ell) \cup (G', \sigma')$ satisfies (K3), we must show that (i) (G_ℓ, σ_ℓ) and (G', σ') are *hc*-cographs, and (ii) $\sigma(V_\ell) \cap \sigma(V') \in \{\sigma(V_\ell), \sigma(V')\}$.

(i) By Lemma 5.43, each connected component $(G_1, \sigma_1), \dots, (G_k, \sigma_k)$ is an *hc*-cograph. Thus, in particular, (G_ℓ, σ_ℓ) is an *hc*-cograph. Furthermore, Thm. 5.9 implies that each connected component $(G_1, \sigma_1), \dots, (G_k, \sigma_k)$ is a co-RBMG. By Thm. 5.3, the latter two arguments imply that (G', σ') is a co-RBMG. Moreover, Thm. 5.3 implies that there exists $1 \leq j \leq k$ such that $\sigma(V_j) = \sigma(V)$ and $j \neq l$ since $\sigma(V_l)$ is minimal w.r.t. inclusion. Hence, we can conclude from Thm. 5.9 that (G', σ') is an *hc*-cograph.

(ii) By Thm. 5.9, (G, σ) is an RBMG. Applying Cor. 5.4, we can conclude that (G, σ) contains a connected component (G^*, σ^*) with $\sigma(V(G^*)) = \sigma(V)$.

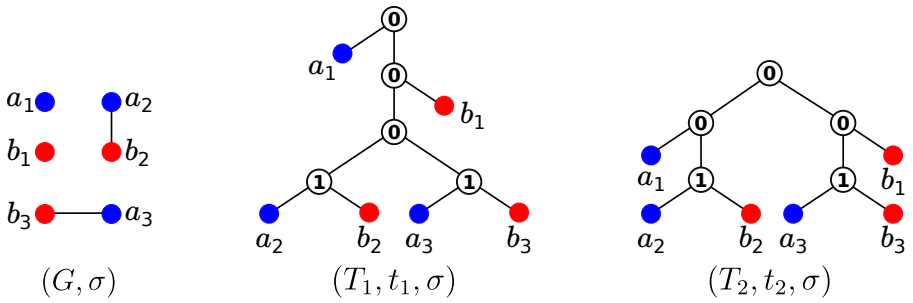


Fig. 30. The graph (G, σ) is an *hc*-cograph and, by Thm. 5.9, a co-RBMG. The trees (T_1, t_1, σ) and (T_2, t_2, σ) correspond to two possible cotrees (T_{hc}^G, t_i, σ) that explain (G, σ). The inner labels "0" and "1" in the cotrees correspond to the values of the maps $t_i : V^0 \to \{0, 1\}$, $i = 1, 2$, such that $xy \in E(G)$ if and only if $t(\mathrm{lca}(x, y)) = 1$. Let $\mathcal{G}_x = (\{x\}, \emptyset), \sigma_x)$ be the colored single vertex graph K_1 with $\sigma_x(x) = \sigma(x)$ for each $x \in \{a_1, a_2, a_3, b_1, b_2, b_3\}$ as indicated in the figure. The tree (T_1, t_1, σ) is constructed based on the valid *hc*-cograph representation $G = \mathcal{G}_{a_1} \cup (\mathcal{G}_{b_1} \cup ((\mathcal{G}_{a_2} \nabla \mathcal{G}_{b_2}) \cup (\mathcal{G}_{a_3} \nabla \mathcal{G}_{b_3})))$. Here, $\mathcal{G}_{a_1}$ plays the role of $\mathcal{G}_\ell$ as in Lemma 5.44. The tree (T_2, t_2, σ) is constructed based on the valid *hc*-cograph representation $G = (\mathcal{G}_{a_1} \cup (\mathcal{G}_{a_2} \nabla \mathcal{G}_{b_2})) \cup (\mathcal{G}_{b_1} \cup (\mathcal{G}_{a_3} \nabla \mathcal{G}_{b_3}))$.

Since $\sigma(V_\ell)$ is minimal w.r.t. inclusion, we can w.l.o.g. assume that (G^*, σ^*) is contained in (G', σ'). Hence, $\sigma(V_\ell) \cap \sigma(V') = \sigma(V_\ell) \cap \sigma(V) = \sigma(V_\ell)$ and thus, $\sigma(V_\ell) \subseteq \sigma(V')$, which implies that $(G, \sigma) = (G_\ell, \sigma_\ell) \cup (G', \sigma')$ satisfies Property (K3cc). □

The choice of the graph G_ℓ in Lemma 5.44 will in general not be unique. As a consequence, there may be distinct cotrees (T_{hc}^G, σ) that explain the same co-RBMG, see Fig. 30 for an illustrative example.

While Thm. 5.8 allows the recognition of co-RBMGs in polynomial time, it does not provide an explaining tree. The equivalence of co-RBMGs and *hc*-cographs together with Lemmas 5.41 and 5.44 yields an alternative polynomial time recognition algorithm that is constructive in the sense that it explicitly provides a tree explaining (G, σ).

Theorem 5.11. *Let (G, σ) be a properly colored undirected graph. Then it can be decided in polynomial time whether (G, σ) is a co-RBMG and, in the positive case, a tree (T, σ) that explains (G, σ) can be constructed in polynomial time.*

Proof. Let $\overline{G}$ denote the complement of G. Testing if (G, σ) is the join or disjoint union of graphs can clearly be done in polynomial time.

Assume first that (G, σ) is the join of graphs. In this case, $(\overline{G}, \sigma)$ decomposes into connected components $(\overline{G}_1, \sigma_1), \ldots, (\overline{G}_k, \sigma_k)$, $k \geq 2$, i.e., $(\overline{G}, \sigma) = \bigcup_{i=1}^k (\overline{G}_i, \sigma_i)$. Therefore $(G, \sigma) = (\overline{\overline{G}}, \sigma) = \overline{\bigcup_{i=1}^k (\overline{G}_i, \sigma_i)} = \nabla_{i=1}^k (\overline{G}_i, \sigma_i) = \nabla_{i=1}^k (G_i, \sigma_i)$, where none of the graphs (G_i, σ_i) can be written as the join of two graphs and k is maximal. Note that we can ignore the parenthesis in the latter equation since the ∇ operation is associative. It follows from (K2cc) that (G, σ) is an *hc*-cograph if and only if (1) all (G_i, σ_i) are *hc*-cographs and (2) all color sets $\sigma(V(G_i))$ are pairwise disjoint. In this case, every binary tree with leaves $(G_1, \sigma_1), \ldots, (G_k, \sigma_k)$ and all inner vertices labeled 1 may appear in (T_{hc}^G, t).

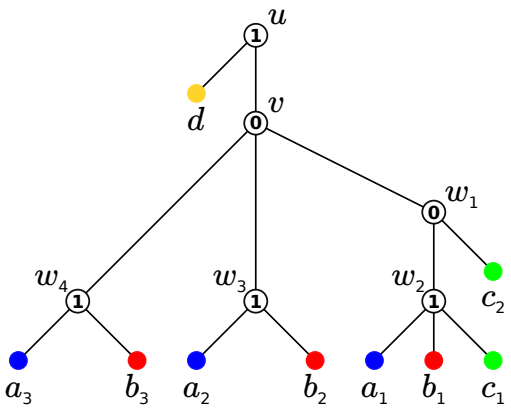


Fig. 31. The cotree (T, t, σ) explains a 4-colored co-RBMG (G, σ). Cor. 5.14 implies that the edge uv is redundant. However, by Prop. 3.1, the tree (T_e, t') is not a cotree for the co-RBMG (G, σ) for all possible labelings $t' : V^0(T) \to \{0, 1\}$. Moreover, Lemma 5.45(2) implies that the two inner edges vw_3 and vw_4 of T are redundant. However, contracting both edges at the same time gives a tree (T_1, σ) with $\mathsf{par}(a_2) = \mathsf{par}(a_3)$, thus a_2 and a_3 belong to the same S-class in $G(T_1, \sigma)$. Hence, (T_1, σ) does not explain (G, σ) (cf. Lemma 5.47). Finally, one checks that the edge vw_1 is relevant because the edge a_3c_2 is contained in $G(T_2, \sigma)$, where (T_2, σ) is obtained from (T, σ) by contraction of vw_1, but not in (G, σ) (cf. Lemma 5.45).

Now assume that (G, σ) is the disjoint union of the connected graphs (G_i, σ_i), $1 \leq i \leq k$. Let $G_\ell = (V_\ell, E_\ell)$, $1 \leq \ell \leq k$, be a connected component such that $\sigma(V_\ell)$ is minimal w.r.t. inclusion. Such a component can be clearly identified in polynomial time. By Lemma 5.44, $(G, \sigma) = (G_\ell, \sigma_{|V_\ell}) \cup (G - G_\ell, \sigma_{|V \setminus V_\ell})$ satisfies (K3), whenever (G, σ) is a co-RBMG. Again, Lemma 5.41 implies that the two trees that explains $(G_\ell, \sigma_{|V_\ell})$ and $(G - G_\ell, \sigma_{|V \setminus V_\ell})$, respectively, can then be joined under a common root in order to obtain a tree that explains (G, σ). The effort is again polynomial.

Finally, each of the latter steps must be repeated recursively on the connected components of either (G, σ) or $(\overline{G}, \sigma)$. This either results in an *hc*-cotree (T^G_{hc}, t, σ) or we encounter a violation of (K2cc) or (K3cc) on the way. That is, we obtain a join decomposition such that the color sets $\sigma(V(G_i))$ are not pairwise disjoint, or a graph G_ℓ such that $(G_\ell, \sigma_{|V_\ell}) \cup (G - G_\ell, \sigma_{|V \setminus V_\ell})$ violates (K3cc). In either case, the recursion terminates and reports "(G, σ) is not an *hc*-cograph". Since T^G_{hc} has $O(|V(G)|)$ vertices, the polynomial time decomposition steps must be repeated at most $O(|V(G)|)$ times, resulting in an overall polynomial time algorithm.

□

Recall that T_e denotes the tree that is obtained from T by contraction of the edge e and (T, t, σ) or (T, t) is a cotree for (G, σ) if $t(\mathrm{lca}(x, y)) = 1$ if and only if $xy \in E(G)$.

Fig. 19 gives an example of a least resolved tree (T, σ) that explains a co-RBMG (G, σ). However, one easily verifies that (T, σ) is not a refinement of the discriminating cotree for (G, σ). Moreover, as shown in Fig. 30, there might exist several cotrees that explain a given co-RBMG. From Prop. 3.1 we

can infer that the discriminating cotree for a co-RBMG (G,σ) is unique. It does not necessarily explain (G,σ), however. In order to see this, consider the example in Fig. 31, where the edge vw_1 with $t(v) = t(w_1) = 0$ cannot be contracted without violating the property that the resulting tree still explains the underlying co-RBMG.

To shed some light on the question how cotrees for a co-RBMG (G,σ) and least resolved trees that explain (G,σ) are related, we identify the edges of a cotree for (G,σ) that can be contracted. To this end, we show that the sufficient conditions in Lemma 5.2 are also necessary for co-RBMGs.

Lemma 5.45. *Let (T,t,σ) be a not necessarily binary cotree explaining the co-RBMG (G,σ) that is also a cotree for (G,σ) and let $e = uv$ be an inner edge of T. Then (T_e,σ) explains (G,σ) if and only if either Property (1) or (2) from Lemma 5.2 is satisfied.*

Proof. By Lemma 5.2, Properties (1) and (2) ensure that (T_e,σ) explains (G,σ).

Conversely, suppose that (T_e,σ) explains (G,σ). Since (G,σ) is a co-RBMG, it is an *hc*-cograph by Thm. 5.9 and thus, (T,t,σ) is an *hc*-cotree for (G,σ). Let $\mathcal{G}_x := (G,\sigma)[L(T(x))]$ for $x \in V(T)$. Clearly, $(T(u), t_{|L(T(u))}, \sigma_{|L(T(u))})$ is an *hc*-cotree for $\mathcal{G}_u$ and thus, $\mathcal{G}_u$ is an *hc*-cograph. Hence, $\mathcal{G}_u$ can be written as $\mathcal{G}_v \star (H,\sigma_H)$, where $\star \in \{\nabla, \cup\}$ and $(H,\sigma_H) = \star_{x\in C}\mathcal{G}_x$ for $C := \mathsf{child}_T(u) \setminus \{v\}$. Clearly, either $t(u) = 1$ (in case $\star = \nabla$) or $t(u) = 0$ (in case $\star = \cup$). As $(T(u), t_{|L(T(u))}, \sigma_{|L(T(u))})$ is an *hc*-cotree, we have either $\sigma(L(T(v'))) \cap \sigma(L(T(v))) = \emptyset$ for all $v' \in \mathsf{child}_T(u)$ (in case $\star = \nabla$) or $\sigma(L(T(v'))) \cap \sigma(L(T(v))) \in \{\sigma(L(T(v))), \sigma(L(T(v')))\}$ for all $v' \in \mathsf{child}_T(u)$ (in case $\star = \cup$).

Thus, if $\star = \nabla$, then we immediately obtain Property (1). In the second case, where $\mathcal{G}_u = \mathcal{G}_v \cup (H', \sigma_{H'})$, the color constraint (K3cc) implies $\sigma(L(T(v))) \subseteq \sigma(L(T(v')))$ or $\sigma(L(T(v'))) \subsetneq \sigma(L(T(v)))$ for any $v' \in \mathsf{child}_T(u)$. If the first case is true for all children of u in T, we obtain Property (2.i) of Lemma 5.2. Thus suppose there exists some vertex $v' \in \mathsf{child}_T(u)$ with $\sigma(L(T(v'))) \subsetneq \sigma(L(T(v)))$. Assume, for contradiction, that there is a vertex $w \in \mathsf{child}_T(v)$ with $S_{w,\neg v'} = \sigma(L(T(x))) \setminus \sigma(L(T(y))) \neq \emptyset$ and a color s such that s is contained in $\sigma(L(T(v')))$ but not in $\sigma(L(T(w)))$. Thus, in particular, there exists a vertex $b \preceq_T v'$ with $\sigma(b) = s$. Moreover, there is vertex $a \preceq_T w$ with $\sigma(a) = r \in S_{w,\neg v'}$. Since $r \notin \sigma(L(T(v')))$, the leaf a must be contained in the out-neighborhood of b in $\vec{G}(T,\sigma)$. Since $\sigma(L(T(v'))) \subsetneq \sigma(L(T(v)))$ and $s \notin \sigma(L(T(w)))$, there exists a vertex $b' \in L(T(v)) \setminus L(T(w))$ with $\sigma(b') = s$. Hence, $\mathrm{lca}(a,b') = v$. Thus b is not contained in the out-neighborhood of a, i.e., $ab \notin E(G)$. However, if we contract $e = uv$, we obtain the new vertex $uv = \mathrm{lca}_{T_e}(a,b)$ in T_e. Since $r \notin \sigma(L(T(v')))$ and $s \notin \sigma(L(T(w)))$, we immediately obtain $\mathrm{lca}_{T_e}(a,b) \preceq_{T_e} \mathrm{lca}_{T_e}(a,b')$ and $\mathrm{lca}_{T_e}(a,b) \preceq_{T_e} \mathrm{lca}_{T_e}(a',b)$ for all a' of color $\sigma(a)$ and b' of color $\sigma(b)$. Thus $ab \in E(G(T_e,\sigma))$ and hence, (T_e,σ) does not explain (G,σ); a contradiction.

□

As an immediate consequence of Lemma 5.45(1) we obtain

Corollary 5.14. *Let (T, t, σ) be a not necessarily binary cotree explaining the co-RBMG (G, σ) that is also a cotree for (G, σ). If $t(u) = 1$ for an inner edge $e = uv$ of T, then the tree (T_e, σ) explains (G, σ).*

Now, Cor. 5.14 and Prop. 3.1 imply

Corollary 5.15. *Let (T, t, σ) be a not necessarily binary cotree explaining the co-RBMG (G, σ) and let $e = uv$ be an inner edge of T with $t(u) = t(v) = 1$. Then the tree (T_e, t_e, σ) explains (G, σ) and is a cotree for (G, σ), where the vertex $w = uv$ obtained by contracting the edge uv is labeled by $t_e(w) = 1$ and $t_e(w') = t(w')$ for all other vertices $w' \neq w$.*

Thus, if (T_{hc}^G, t, σ) is a least resolved tree that explains (G, σ), then it will not have any adjacent vertices labeled by 1. The situation is more complicated for 0-labeled vertices. Fig. 31 shows that not all edges uv in (T_{hc}^G, t, σ) with $t(u) = t(v) = 0$ can be contracted. However, we obtain the following characterization, which is an immediate consequence of Prop. 3.1, Lemma 5.45, and Cor. 5.15.

Corollary 5.16. *Let (T, t, σ) be a not necessarily binary cotree explaining the co-RBMG (G, σ) that is also a cotree for (G, σ). Let $e = uv$ be an inner edge of T. The following two statements are equivalent:*

1. *(T_e, t_e, σ) explains (G, σ) and is a cotree for (G, σ), where the vertex $w = uv$ obtained by contracting the edge uv is labeled by $t_e(w) = t(u)$ and $t_e(w') = t(w')$ for all other vertices $w' \neq w$,*
2. *$t(u) = t(v)$ and, if $t(u) = 0$, then e satisfies Properties (1) and (2) in Lemma 5.2.*

If we apply Cor. 5.15 and 5.16, then Prop. 3.1 implies that we always obtain a cotree (T_e, t_e, σ) for (G, σ). Hence, we can repeatedly apply Cor. 5.15 and 5.16 and conclude that the least resolved tree (T, t, σ) explaining (G, σ) does neither contain edges uv with $t(u) = t(v) = 1$ nor edges uv with $t(u) = t(v) = 0$ satisfying Lemma 5.2(1) and (2). Moreover, Cor. 5.14 allows us to contract edges uv with $t(u) = 1 \neq t(v)$. In this case, however, Prop. 3.1 implies that (T_e, t') is not a cotree for the co-RBMG (G, σ) for all possible labelings $t' : V^0 \to \{0, 1\}$. Hence, the question arises how often we can apply Cor. 5.14. An answer is provided by the next result:

Lemma 5.46. *Let (T, t, σ) be a not necessarily binary cotree that explains the co-RBMG (G, σ) and that is a cotree for (G, σ). Let A be the set of all inner edges $e = uv$ of T with $t(u) = 1$. Then (T_B, σ) explains (G, σ) for all $B \subseteq A$.*

Proof. Any edge $e = uv \in B$ is contracted to some vertex u_e in (T_B, σ).

Let $e = uv \in A$. By definition of the set A, we have $t(u) = 1$. Clearly, $(T(u), t_{|L(T(u))}, \sigma_{|L(T(u))})$ is an *hc*-cotree. Hence, $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) = \emptyset$ for any two distinct vertices $v_1, v_2 \in \mathsf{child}_T(u)$. Now assume that we have contracted e to obtain (T_e, σ). By Cor. 5.14, (T_e, σ) explains (G, σ). Moreover, suppose that there exists another edge $f = uv' \in A$ which corresponds to $u_e v'$ in (T_e, σ). In (T, σ), we have $\sigma(L(T(v))) \cap \sigma(L(T(v'))) = \emptyset$ which, in

particular, implies $\sigma(L(T(v'))) \cap \sigma(L(T(w))) = \emptyset$ for all $w \in \mathsf{child}_T(v)$. In (T_e, σ), the children of u_e are now $\mathsf{child}_{T_e}(u_e) = (\mathsf{child}_T(u) \setminus \{v\}) \cup \mathsf{child}_T(v)$. Thus $\sigma(L(T(v'))) \cap \sigma(L(T(v''))) = \emptyset$ for all $v'' \in \mathsf{child}_{T_e}(u_e)$. Lemma 5.2(1) implies that f can be contracted to obtain the tree (T_{ef}, σ) that explains (G, σ). Repeated application of the latter arguments shows that all edges incident to vertex u in (T, σ) can be contracted.

Finally, the contraction of the edges can be performed in a top-down fashion. In this case, the contraction of edges incident to u does not influence the children of any vertex u' that is incident to some edge $e' = u'v'$ having label $t(u') = 1$. That is, we can apply the latter arguments to all edges in B independently, from which we conclude that (T_B, σ) explains (G, σ) for all $B \subseteq A$.

□

For the contraction of edges uv with $t(u) = 0 \neq t(v)$, however, the situation becomes more complicated.

Lemma 5.47. *Let (T, t, σ) be a not necessarily binary cotree that explains the co-RBMG (G, σ) and is a cotree for (G, σ). Moreover, let $u \in V^0(T)$ be an inner vertex with $t(u) = 0$ and $A = \{e_1, \ldots, e_k\}$ be the set of all inner edges $e_i = uv_i \in E(T)$ with $v_i \in \mathsf{child}_T(u)$ such that $t(v_i) = 1$ and e_i is redundant in (T, σ). Then (T_e, σ) explains (G, σ) for all $e \in A$ but (T_B, σ) does not explain (G, σ) for all $B \subseteq A$ with $|B| \geq 2$.*

Proof. Let the edges $e = uv$ and $f = uv'$ be contained in A. Since e and f are redundant, (T_e, σ) and (T_f, σ) both explain (G, σ). It suffices to show that (T_{ef}, σ) does not explain (G, σ). Following the same argumentation as in the beginning of the proof of Lemma 5.45, we conclude that $(T(v), t_{|L(T(v))}, \sigma_{|L(T(v))})$ is an hc-cotree. This and $t(v) = 1$ implies $\sigma(L(T(w_i))) \cap \sigma(L(T(w_j))) = \emptyset$ for any two distinct vertices $w_i, w_j \in \mathsf{child}_T(v)$. Hence, $\sigma(L(T(v)))$ is partitioned into the sets $\sigma(L(T(w_1))), \ldots, \sigma(L(T(w_k)))$ with $w_i \in \mathsf{child}_T(v)$, $1 \leq i \leq k$. Analogously, $\sigma(L(T(w'_1))), \ldots, \sigma(L(T(w'_m)))$ with $w'_i \in \mathsf{child}_T(v')$, $1 \leq i \leq m$ forms a partition of $\sigma(L(T(v')))$. Consider an arbitrary but fixed vertex $w \in \mathsf{child}_T(v)$. Assume, for contradiction, that (T_{ef}, σ) explains (G, σ) and denote by u_{ef} the inner vertex in T_{ef} that is obtained by contracting the edges uv and uv'. Since (G, σ) is an hc-cograph and $t(u) = 0$, the color sets $\sigma(L(T(v)))$ and $\sigma(L(T(v')))$ are neither disjoint nor do they overlap. As $(T_{ef}, \sigma) = (T_{fe}, \sigma)$, we can w.l.o.g. assume $\sigma(L(T(v'))) \subseteq \sigma(L(T(v)))$.

Now let $w' \in \mathsf{child}_T(v')$ such that there is some $z' \in L(T(w'))$ with $\sigma(z') = t \notin \sigma(L(T(w)))$. Let $x \in L(T(w))$ with $\sigma(x) = r$. Since $t(u) = 0$ and $u = \mathrm{lca}_T(x, z')$, we have $xz' \notin E(G)$. However, as $\sigma(L(T(v'))) \subseteq \sigma(L(T(v)))$, there is some child $\widetilde{w} \in \mathsf{child}_T(v)$ distinct from w such that $t \in \sigma(L(T(\widetilde{w})))$. Let $\widetilde{z} \in L(T(\widetilde{w}))$ with $\sigma(\widetilde{z}) = t$. Since $t(v) = 1$, we have $x\widetilde{z} \in E(G)$. As (T_{ef}, σ) explains (G, σ), $x\widetilde{z}$ must be an edge in $G(T_{ef}, \sigma)$. Hence, $\mathrm{lca}_{T_{ef}}(x, \widetilde{z}) = u_{ef} \preceq_{T_{ef}} \mathrm{lca}_{T_{ef}}(x, z'')$ and $\mathrm{lca}_{T_{ef}}(x, \widetilde{z}) = u_{ef} \preceq_{T_{ef}} \mathrm{lca}_{T_{ef}}(x'', \widetilde{z})$ for all $x'' \in L[\sigma(x)]$ and $z'' \in L[t]$. However, by construction of T_{ef}, we have $\mathrm{lca}_{T_{ef}}(x, z') = \mathrm{lca}_{T_{ef}}(x, \widetilde{z}) = u_{ef}$. Hence, if $r \notin \sigma(L(T(w')))$, then $x\widetilde{z} \in E(G(T_{ef}, \sigma))$ implies that xz' is an edge in $G(T_{ef}, \sigma)$; a contradiction to $xz' \notin E(G)$. Now assume $r \in \sigma(L(T(w')))$. Clearly, v' must have at least one further child w'' with $s \in \sigma(L(T(w'')))$ and $s \notin \sigma(L(T(w')))$. In particular, $r, t \notin \sigma(L(T(w'')))$.

Algorithm 5 From Cotrees to Least Resolved Trees

Require: Leaf-labeled cotree (T, t, σ)

1: $A \leftarrow \emptyset$
2: **for all** inner edges $e = uv$ with $t(u) = 1$ **do**
3: $\quad A \leftarrow A \cup \{e\}$
4: $(T, \sigma) \leftarrow (T_A, \sigma)$
5: **while** (T, σ) contains redundant inner edges $e = uv$ **do**
6: $\quad$ Contract e to obtain (T_e, σ)
7: $\quad (T, \sigma) \leftarrow (T_e, \sigma)$
8: **return** (T, σ)

Since $\sigma(L(T(v'))) \subseteq \sigma(L(T(v)))$, there exists a leaf $y \in L(T(v))$ with $\sigma(y) = s$. Now we either have $y \preceq_T w$ or $y \preceq_T \widetilde{w}$ or $y \preceq_T \hat{w} \in \mathsf{child}_T(v) \setminus \{w, \widetilde{w}\}$. In any case, $t(v) = 1$ implies that at least one of the edges xy or $y\widetilde{z}$ must be contained in G. Assume $xy \in E(G)$. Since $t(u) = 0$, we have $xy'' \notin E(G)$ for any $y'' \in L(T(w'')) \cap L[s]$. On the other hand, as $r \notin \sigma(L(T(w'')))$, we can apply the preceding argumentation to infer from $xy \in E(G(T_{ef}, \sigma))$ that xy'' is an edge in $G(T_{ef}, \sigma)$; a contradiction. Analogously, the existence of an edge $y\widetilde{z}$ in G yields a contradiction as well. Hence, (T_{ef}, σ) does not explain (G, σ). □

The previous results can finally be used to obtain a least resolved tree (T, σ) from a cotree (T', t, σ) for a given co-RBMG (G, σ) that also explains (G, σ). Instead of checking all inner edges of (T', σ) for redundancy, Lemma 5.46 can be applied to identify promptly many redundant edges, which considerably reduces the number of edges that need to be checked. This idea is implemented in Algorithm 5, which returns a least resolved tree explaining (G, σ). Lemma 5.47, however, suggests that this least resolved tree is not necessarily unique for (G, σ).

Theorem 5.12. *Let (G, σ) be a co-RBMG that is explained by (T, t, σ) such that (T, t, σ) is also a cotree for (G, σ). Then Algorithm 5 returns a least resolved tree that explains (G, σ), in polynomial time.*

Proof. Lemma 5.46 implies that all inner edges $e = uv$ with $t(u) = 1$ can be contracted, which is done in Line 2-4. Afterwards we check for all remaining inner edges $e = uv$ whether they are redundant or not and, if so, contract them. In summary, the algorithm is correct. Clearly, all steps including the check for redundancy as in Lemma 5.1 can be done in polynomial time. □

5.8 SUMMARY

Reciprocal best match graphs are the symmetric parts of best match graphs (see Chapter 4). They have a surprisingly complicated structure that makes it quite difficult to recognize them. Although we have succeeded here in obtaining a complete characterization of 3-RBMGs, it remains an open problem whether general n-RBMGs can be recognized in polynomial time. This is in striking

contrast to the directed BMGs, which are recognizable in polynomial time. The key difference between the directed and symmetric version is that every BMG $(\vec{G}, \sigma)$ is explained by a unique least resolved tree which is displayed by every tree that explains $(\vec{G}, \sigma)$. RBMGs, in contrast, can be explained by multiple, mutually inconsistent trees. This ambiguity seems to be the root cause of the complications that are encountered in the context of RBMGs with more than three colors.

An important subclass of RBMGs are the ones that have cograph structure (co-RBMGs). These are good candidates for correct estimates of the orthology relation. Interestingly, they are easy to recognize: by Thm. 5.8 it suffices to check that all connected 3-colored restrictions of an RBMG are cographs. Moreover, hierarchically colored cographs (*hc*-cographs) characterize co-RBMGs. Thm. 5.11 shows that co-RBMGs (G, σ) can be recognized in polynomial time. In addition, Thm. 5.11 and 5.12 imply that a least resolved tree that explains (G, σ) can be constructed in polynomial time. Since each orthology relation is equivalently represented by a cograph, every co-RBMG (G, σ) represents an orthology relation. The converse, however, is not always satisfied, as not all mathematically valid orthology relations are *hc*-cographs. The relationships of orthology relations and RBMGs will be the topic of the next chapter.

6

BEST MATCH GRAPHS AND RECONCILIATION OF GENE TREES WITH SPECIES TREES

Despite its practical importance, the mathematical interrelationships of best matches on one hand, and reconciliations of gene and species trees on the other hand have remained largely unexplored. The purpose of this chapter is to bridge that gap between (reciprocal) best matches and reconciliation maps. While it is true that any gene tree, and thus also any best match graph, can be reconciled with any species tree [79], such a reconciliation may imply unrealistically many duplication and deletion events. Although orthology implies a cograph structure, it is not necessarily true that reciprocal best matches form a cograph. One of the main result in this chapter shows, however, that, in the absence of HGT, the true orthology graph (TOG) is a subgraph of the reciprocal best match graph (Section 6.2). Furthermore, conditions under which a co-RBMG identifies the correct orthology relation are established in Sections 6.3 and 6.4. Computer simulations in Section 6.5 show that in a broad parameter range the TOG and RBMG are very similar, proving an *a posteriori* justification for the use of reciprocal best matches in orthology estimation. Moreover, these simulations reveal that most false positive orthology assignments can be identified as good quartets – and thus corrected – in the absence of horizontal transfer. Horizontal transfer, however, may introduce also false negative orthology assignments. The chapter is started with some important prerequisites and findings about the reconciliation map and the event labeling.
This chapter is based on the results in Geiß et al. [74].

6.1 RECONCILIATION MAP AND EVENT LABELING

This section makes intensive use of planted *phylogenetic trees* (see Section 3.3.1 for a definition). The main reason for using planted phylogenetic trees instead of modeling phylogenetic trees simply as rooted trees, which is the much more common practice in the field, is that we will often need to refer to the time before the first branching event. Conceptually, this corresponds to explicitly representing an outgroup. Whenever not stated otherwise, the trees in this chapter are assumed to be planted phylogenetic trees.

Let a gene tree $T = (V, E)$ and a species tree $S = (W, F)$ be planted phylogenetics trees on a set of (extant) genes $L(T)$ and species $L(S)$, respectively. Their planted roots will be denoted by 0_T and 0_S, resp., and their conventional roots by ρ_T and ρ_S. Recall that the only neighbor of a planted root is the conventional root and it is neither contained in the set of leaves nor in the set of inner vertices. We assume that the map $\sigma\colon L(T) \to L(S)$ that assigns to each gene the species in whose genome it resides, is known. Recall from the introductory Chapter 2 that a reconciliation of T and S is an extension of this map which maps the ancestors of the extant genes into the species tree:

Definition 6.1. *Let $S = (W, F)$ and $T = (V, E)$ 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 $\mu\colon V \to W \cup F$ 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. *$x \prec_T y$ implies $\mu(x) \preceq_S \mu(y)$.*

(R3) Speciation Constraints. *Suppose $\mu(x) \in W^0$.*

(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.

The axiom system above is equivalent to the following version, which commonly has been used in the literature, see e.g. Górecki and Tiuryn [79], Vernot et al. [227], Doyon et al. [57], Rusin et al. [194], Hellmuth [91], Nøjgaard et al. [173], and the references therein:

Lemma 6.1. *Let μ be a map from $(T = (V, E), \sigma)$ to $S = (W, F)$ that satisfies* (R0) *and* (R1)*. Then μ satisfies Axioms* (R2) *and* (R3) *if and only if μ satisfies*

(R2') Ancestor Constraint.
Suppose $x, y \in V$ with $x \prec_T y$.

(i) If $\mu(x), \mu(y) \in F$, then $\mu(x) \preceq_S \mu(y)$,

(ii) otherwise, i.e., at least one of $\mu(x)$ and $\mu(y)$ is contained in W, $\mu(x) \prec_S \mu(y)$.

(R3') Inner Vertex Constraint.
If $\mu(x) \in W^0$, then

(i) $\mu(x) = \mathrm{lca}_S(\sigma(L(T(x))))$ and

(ii) $\mu(v')$ and $\mu(v'')$ are incomparable in S for any two distinct children v' and v'' of x in T.

Proof. Assume first that (R2) and (R3) are satisfied for μ.
Then Property (R2'.i) is satisfied since it is the restriction of (R2) to $\mu(x), \mu(y) \in F$.
In order to see that (R2'.ii) holds, let $x \prec_T y$ and $\mu(x) \in W$ or $\mu(y) \in W$. Assume first $\mu(y) \in W$. Property (R2) implies $\mu(x) \preceq_S \mu(y)$. Let v be the child of y that lies on the path from y to x in T, i.e., $x \preceq_T v \prec_T y$. Assume, for contradiction, that $\mu(x) = \mu(y)$. By Property (R2), we have $\mu(x) = \mu(v) = \mu(y)$. For every other child v' of y, Property (R2) implies $\mu(v') \preceq_S \mu(y) = \mu(v)$. Thus $\mu(v)$ and $\mu(v')$ are comparable; a contradiction to (R3.ii). Hence, $\mu(x) \prec_S \mu(y)$ and (R2'.ii) is satisfied. Now suppose $\mu(x) \in W$ and assume, for contradiction, that $\mu(x) = \mu(y)$. Thus $\mu(y) \in W$ and we can apply the same arguments as above to conclude that (R3.ii) is not satisfied. Hence, $\mu(x) \prec_S \mu(y)$ and (R2'.ii)

is satisfied.
In order to show that (R3') is satisfied, let $x \in V$ such that $\mu(x) \in W^0$. Properties (R3'.ii) and (R3.ii) are equivalent. It remains to show that (R3'.i) is satisfied. From (R2) we infer $\mu(y) \preceq_S \mu(x)$ for all $y \in \bigcup_{v \in \mathsf{child}(x)} L(T(v)) = L(T(x))$. Thus

$$\mu(x) \succeq_S \mathrm{lca}_S(\sigma(L(T(x)))). \tag{19}$$

Property (R3.i) implies that there are two distinct children $v', v'' \in \mathsf{child}(x)$ with $\mu(x) = \mathrm{lca}_S(\mu(v'), \mu(v''))$. Again using (R3.ii), we know that the images $\mu(v')$ and $\mu(v'')$ are incomparable in S. The latter together with $\mu(y) \preceq_S \mu(v')$ for all $y \in L(T(v'))$ and $\mu(y') \preceq_S \mu(v'')$ for all $y' \in L(T(v''))$ implies

$$\mathrm{lca}_S(\mu(v'), \mu(v'')) = \mathrm{lca}_S(\sigma(L(T(v'))) \cup \sigma(L(T(v'')))) \preceq_S \mathrm{lca}_S(\sigma(L(T(x)))).$$

In summary, $\mathrm{lca}_S(\sigma(L(T(x)))) \preceq_S \mu(x) = \mathrm{lca}_S(\mu(v'), \mu(v'')) \preceq_S \mathrm{lca}_S(\sigma(L(T(x))))$ implies that $\mu(x) = \mathrm{lca}_S(\sigma(L(T(x))))$ and Property (R3'.i) is satisfied.
Therefore (R2) and (R3) imply (R2') and (R3').

Conversely, assume that (R2') and (R3') are satisfied for μ. Clearly, (R2') implies (R2), and (R3'.ii) implies (R3.ii). It remains to show that (R3.i) is satisfied. Let $\mu(x) \in W^0$. By (R2'.ii), we have $\mu(x) \succ_S \mu(v_i)$ for all children $v_i \in \mathsf{child}(x) = \{v_1, \ldots, v_k\}$, $k \geq 2$. Therefore $\mu(x) \succeq_S \mathrm{lca}_S(\mu(v_1), \ldots, \mu(v_k))$. By (R3'.ii), the images $\mu(v_1), \ldots, \mu(v_k)$ are pairwise incomparable in S. The latter and (R2'.i) imply $\mathrm{lca}_S(\mu(v_1), \ldots, \mu(v_k)) = \mathrm{lca}_S(\bigcup_{i=1}^k \sigma(L(T(v_i)))) = \mathrm{lca}_S(\sigma(L(T(x)))) = \mu(x)$. It is easy to verify that $\mathrm{lca}_S(\mu(v_1), \ldots, \mu(v_k)) = \mathrm{lca}_S(\mu(v'), \mu(v''))$ for at least two children $v', v'' \in \mathsf{child}(x)$ is always satisfied. Hence, $\mu(x) = \mathrm{lca}_S(\mu(v'), \mu(v''))$ for some $v', v'' \in \mathsf{child}(x)$ and thus, (R3.i) is satisfied.
Therefore (R2') and (R3') imply (R2) and (R3).

□

A simple consequence of the axioms is

Lemma 6.2. *Let μ be a reconciliation map from the leaf-colored tree (T, σ) to $S = (W, F)$ and $x \in V(T)$ a vertex with $\mu(x) \in W^0$. Then $\sigma(L(T(v'))) \cap \sigma(L(T(v''))) = \emptyset$ for all distinct $v', v'' \in \mathsf{child}(x)$.*

Proof. Assume, for contradiction, that there is a vertex $z \in \sigma(L(T(v'))) \cap \sigma(L(T(v'')))$. By Condition (R2'), we have $\mu(x) \succ_S \mu(v') \succeq_S z$ and $\mu(x) \succ_S \mu(v'') \succeq_S z$. Thus there is a path P_1 from $\mu(x)$ to z that contains $\mu(v')$ and a path P_2 from $\mu(x)$ to z that contains $\mu(v'')$. However, Condition (R3.ii) implies that $\mu(v')$ and $\mu(v'')$ are incomparable in S, that is, the subtree of S consisting of the two paths P_1 and P_2 must contain a cycle; a contradiction. □

It is a well-known result that it is always possible to reconcile a given pair of gene tree T and species tree S, see e.g. [79]. For convenience, we include a short direct proof of this fact.

Lemma 6.3. *For every tree $(T = (V, E), \sigma)$ there is a reconciliation map μ to any species tree S with leaf set $L(S) = \sigma(L(T))$.*

Proof. Let $S = (W, F)$ be an arbitrary species tree with leaf set $L(S)$ and $e_0 = 0_S\rho_S$ be the unique root-edge of S. Set $\mu(0_T) = 0_S$ and $\mu(v) = \sigma(v)$ for all $v \in L(T)$. Thus (R0) and (R1) are satisfied. Now set $\mu(v) = e_0$ for all $v \in V^0 = V \setminus (L(T) \cup \{0_T\})$. Thus $\mu(v) \notin W^0$ for all $v \in V^0$ and (R3) is trivially satisfied. Finally, for all $v, v' \in V^0$ and $y \in L(T)$ with $y \prec_T v \prec_T v'$ we have, by construction of μ, $\mu(y) \prec_T \mu(v) = \mu(v') \prec_T \mu(0_T)$. Thus (R2) is satisfied.

□

The reconciliation map also completely determines an *event labeling* on the vertices of T.

Definition 6.2. *Given a reconciliation map μ from (T, σ) to S, the* event labeling on T (determined by μ) *is the map $t_T : V(T) \to \{\circledcirc, \odot, \bullet, \square\}$ given by:*

$$t_T(u) = \begin{cases} \circledcirc & \text{if } u = 0_T, \text{ i.e., } \mu(u) = 0_S \text{ (root)} \\ \odot & \text{if } u \in L(T), \text{ i.e., } \mu(u) \in L(S) \text{ (leaf)} \\ \bullet & \text{if } \mu(u) \in V^0(S) \text{ (speciation)} \\ \square & \text{else, i.e., } \mu(u) \in E(S) \text{ (duplication)} \end{cases}$$

A given gene tree (T, σ) together with a specified map $t_T : V(T) \to \{\circledcirc, \odot, \bullet, \square\}$ is denoted by (T, t_T, σ). One can characterize in polynomial time whether there is reconciliation map μ between a gene tree (T, t_T, σ) and a given species tree that implies t_T [99]. In other words, the analog of Lemma 6.3 is not true if event labels are prescribed at the inner vertices of the gene tree T [105]. Recall from Section 2.1 that two distinct leaves $x, y \in L(T)$ are called *orthologs (w.r.t. μ)* if $t_T(\mathrm{lca}_T(x, y)) = \bullet$ and *paralogs* if $t_T(\mathrm{lca}_T(x, y)) = \square$. For completeness, note that $t_T(\mathrm{lca}_T(x, y)) = \odot$ if and only $x = y$, and 0_T is never the last common ancestor of any of pair of leaves since the planted root 0_T has degree 1 by construction.

We give at this point a more formal definition of the orthology relation based on the event labeling of a gene tree:

Definition 6.3. *Let Θ be a binary relation on L and let T be a planted tree with event map t. Define $\Theta(T, t_T)$ as the set of all pairs (x, y) with $t_T(\mathrm{lca}_T(x, y)) = \bullet$, $x, y \in L(T)$. We say that* Θ is explained *by (T, t_T), if $\Theta = \Theta(T, t_T)$. In this case we call Θ an* orthology relation.

Recall that the orthology relation Θ explicitly depends on the event labeling. In analogy with Def. 6.3, one can also define the *paralogy relation* $\overline{\Theta}$ by $t_T(\mathrm{lca}_T(x, y)) = \square$. Both orthology and paralogy are irreflexive since $t_T(\mathrm{lca}_T(x, x)) = t_T(x) = \odot$ for all leaves $x \in L(T)$. Recall that both relations are symmetric but not transitive (cf. Section 2.1). Note that, in the absence of HGT, orthology Θ and paralogy $\overline{\Theta}$ are complementary in the graph-theoretical sense, i.e., (x, y) – and, by symmetry, also (y, x) – is contained in exactly one of Θ or $\overline{\Theta}$.

Based on the work of Böcker and Dress [21] it has been shown by Hellmuth et al. [97] that Θ is a valid orthology relation, i.e., Θ is explained by an event labeled tree (T, t_T), if and only if Θ is a cograph. Furthermore, a cograph Θ is explained (in the sense of Def. 6.3) by its cotree by replacing the labeling 0 and 1 by $\square$ and $\bullet$, respectively.

According to Lemma 6.3 there is a reconciliation map from (T, σ) to every species tree with leaf set $\sigma(L(T))$. However, this is no longer true when an event labeling is prescribed for T. Given (T, t_T, σ), denote by $\mathcal{S}(T, t_T, \sigma)$ the set of triples $\sigma(a)\sigma(b)|\sigma(c)$ for which $ab|c$ is a triple displayed by T such that (i) $\sigma(a)$, $\sigma(b)$, $\sigma(c)$ are pairwise distinct and (ii) the root of the triple is a speciation event, i.e., $t_T(\mathrm{lca}(a, b, c)) = \bullet$. This set of triples characterizes the existence of a reconciliation map:

Proposition 6.1. *[105, 99] Given an event-labeled, leaf-labeled tree (T, t_T, σ) and species tree S with $L(S) = \sigma(L(T))$, there is a reconciliation map $\mu : V(T) \to V(S) \cup E(S)$ such that the event labeling is consistent with Def. 6.2 if and only if S displays $\mathcal{S}(T, t_T, \sigma)$. In particular, (T, t_T, σ) can be reconciled with a species tree if and only if $\mathcal{S}(T, t_T, \sigma)$ is consistent.*

Clearly, it is possible to find event labelings for T such that there is a reconciliation with any gene tree S. In particular, this is the case whenever $\mathcal{S}(T, t_T, \sigma) = \emptyset$. This in particular holds if (T, t_T, σ) contains no speciation with descendants in three different species or if all children of a speciation vertex are leaves.

Corollary 6.1. *Let (T, t_T, σ) be an event-labeled and leaf-labeled tree. If $|\sigma(L(T(v)))| \leq 2$ for every $v \in V(T)$ with $t_T(v) = \bullet$, then there is a reconciliation map from T to any species tree S.*

It is known that the species triples can directly be obtained from the orthology relation Θ without the need to construct the gene tree [96]. We have $\sigma(a)\sigma(b)|\sigma(c) \in \mathcal{S}(T, t_T, \sigma)$ if and only if $\sigma(a), \sigma(b)$, and $\sigma(c)$ are pairwise different species and either

(a) $(a, c), (b, c) \in \Theta$ and $(a, b) \notin \Theta$ *or*

(b) $(a, c), (b, c), (a, b) \in \Theta$ and there is a vertex $d \neq a, b, c$ with $(c, d) \in \Theta$ and $(a, d), (b, d) \notin \Theta$.

This simple rule in particular applies to co-RBMGs (G, σ). It seems likely that it also generalizes to induced subgraphs (H, σ') of (G, σ) that are cographs. This will be analyzed in forthcoming work.

6.2 ORTHOLOGY AND BEST MATCHES

The main result of this section shows that the true orthology relation is contained in the reciprocal best match graph:

Theorem 6.1. *Let T and S be planted trees, $\sigma : L(T) \to L(S)$ a surjective map, and μ a reconciliation map from (T, σ) to S. If $x, y \in L(T)$ are orthologous w.r.t. (the event map t_T defined by) μ, then x and y are reciprocal best matches in (T, σ).*

Proof. Assume that $x, y \in L(T)$ are orthologous w.r.t. μ and let $u = \mathrm{lca}_T(x, y)$. By definition of orthologs, $\mu(u) \in W^0$ and $t_T(u) = \bullet$. By construction, there are distinct children $v_1, v_2 \in \mathsf{child}(u)$ such that $x \preceq_T v_1$ and $y \preceq_T v_2$. Since $\mu(u) \in W^0$, we can apply Lemma 6.2 to conclude that $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) = \emptyset$. Hence, there are $X, Y \in L(S)$ such that $\sigma(x) = X \neq \sigma(y) = Y$, and, in particular, $Y \notin \sigma(L(T(v_1)))$.

Assume, for contradiction, that y is not a best match of x. Hence, there is a leaf $y' \in L(T)$ with $\sigma(y') = Y$ such that $w = \mathrm{lca}_T(x, y') \prec_T \mathrm{lca}_T(x, y)$. This implies that w must be located on the path from x to v_1 and hence, $y' \preceq_T w \preceq_T v_1$. But then $y' \in L(T(v_1))$ and thus, $Y \in \sigma(L(T(v_1)))$; a contradiction. Hence, y is a best match of x. By similar arguments, x must be a best match of y and therefore, x and y are reciprocal best matches in (T, σ). □

Observation 6.1. *Reciprocal best matches therefore cannot produce false negative orthology assignments as long as the evolution of a gene family proceeds via duplications, losses, and speciations only.*

In practical application we usually do not know the event-labeled gene tree. It is possible, however, to compute the reciprocal best matches directly from sequence data. Therefore it is of interest to investigate the relationship of colored best match graphs and orthology relations.

Every tree (T, σ) explains a reciprocal best match graph $G(T, \sigma)$. On the other hand, we can endow (T, σ) with a special event labeling $\hat{t}_T : V(T) \to \{\circledcirc, \odot, \bullet, \square\}$, which, motivated by Lemma 6.2, maximizes the number of speciations. Our interest is then to understand the constraints imposed by $(T, \hat{t}_T, \sigma)$.

Definition 6.4. *(T, σ) be a leaf-labeled tree explaining the RBMG (G, σ). The* extremal event labeling *of T is the map $\hat{t}_T : V(T) \to \{\circledcirc, \odot, \bullet, \square\}$ defined for $u \in V(T)$ by*

$$\hat{t}_T(u) = \begin{cases} \circledcirc & \text{if } u = 0_T \\ \odot & \text{if } u \in L(T) \\ \square & \text{if there are two children } u_1, u_2 \in \mathsf{child}(u) \text{ such that} \\ & \qquad \sigma(L(T(u_1))) \cap \sigma(L(T(u_2))) \neq \emptyset \\ \bullet & \text{otherwise} \end{cases}$$

An analogous result as in Thm. 6.1 holds for the extremal event labeling:

Lemma 6.4. *If (T, σ) with leaf set L explains the RBMG (G, σ) and $\hat{t}_T$ is the extremal event labeling of (T, σ), then $\Theta(T, \hat{t}_T)$ is a subgraph of the RBMG $G(T, \sigma)$.*

Proof. Consider a vertex $u \in V^0(T)$ with $\mathsf{child}(u) = \{u_1, \ldots, u_k\}$. If $\hat{t}_T(u) = \square$, then none of the edges xy in G with $x \in L(T(u_i))$ and $y \in L(T(u_j))$, $1 \leq i < j \leq k$ is contained in $\Theta(T, \hat{t}_T)$.
Now suppose $\hat{t}_T(u) = \bullet$. For $x \in L(T(u_i))$ and $y \in L(T(u_j))$ with $1 \leq i <$

$j \leq k$, we have $xy \in \Theta(T, \hat{t}_T)$ and, by construction of $\hat{t}_T$, $\sigma(x) \neq \sigma(y)$. In particular, $\hat{t}_T(u) = \bullet$ implies that all distinct children $u_i, u_j \in \mathsf{child}(u)$ satisfy $\sigma(L(T(u_i))) \cap \sigma(L(T(u_j))) = \emptyset$. Thus $\mathrm{lca}_T(x, y) = u \preceq_T \mathrm{lca}_T(x', y)$ for all $x' \neq x$ with $\sigma(x') = \sigma(x)$ and $\mathrm{lca}_T(x, y) = u \preceq_T \mathrm{lca}_T(x, y')$ for all $y' \neq y$ with $\sigma(y') = \sigma(y)$, i.e., x and y are reciprocal best matches. Hence, $xy \in E(G)$ and thus, $\Theta(T, \hat{t}_T) \subseteq G$. □

The following result about least resolved trees w.r.t. some RBMG is tightly linked to the extremal event labeling. Least resolved trees are interesting in this context because they reflect the maximal information that can be inferred from sequence data.

Lemma 6.5. *Let (G, σ) be an RBMG that is explained by (T, σ). If (T, σ) is least resolved w.r.t. (G, σ), then every inner edge $e = uv \in E(T)$ satisfies $\sigma(L(T(v))) \cap \sigma(L(T(u)) \setminus L(T(v))) \neq \emptyset$.*

Proof. For contraposition, assume that there is an inner edge $e = uv \in E(T)$ with $\sigma(L(T(v))) \cap \sigma(L(T(u)) \setminus L(T(v))) = \emptyset$. Hence, for all $x \in L(T(v))$ and $y \in L(T(u)) \setminus L(T(v))$, we have $\mathrm{lca}_T(x, y) = u$ and $\sigma(x) = X \neq \sigma(y) = Y$. It is easy to see that all such x and y form a reciprocal best match and thus, $xy \in E(G)$. Clearly, x and y form also reciprocal best match in (T_e, σ) and thus, each edge $xy \in E(G)$ with $x \in L(T(v))$ and $y \in L(T(u)) \setminus L(T(v))$ is contained in $G(T_e, \sigma)$. Since we have not changed the relative ordering of the lca's of the remaining vertices, all edges in $E(G)$ are contained in $G(T_e, \sigma)$. □

The converse of Lemma 6.5 is not necessarily true. As an example, consider an inner edge $e = uv \in E(T)$ with $\sigma(L(T(u))) = \sigma(L(T(v))) = \{c\}$. It is easy to see that e can be contracted.

Lemma 6.5 implies that, if (T, σ) is least resolved w.r.t. $G(T, \sigma)$ and $u \in V^0(T)$ such that u is incident to some other inner vertex $v \in \mathsf{child}(u)$, then there is a child $v' \neq v$ of u which satisfies $\sigma(L(T(v'))) \cap \sigma(L(T(v))) \neq \emptyset$. By construction of $\hat{t}_T$, we have $\hat{t}_T(u) = \square$. The latter observation also implies the following:

Corollary 6.2. *Suppose that (T, σ) is least resolved w.r.t. $G(T, \sigma)$ and let $\hat{t}_T$ be the extremal event labeling for (T, σ). Then $\hat{t}_T(u) = \bullet$ if and only if all children of u are leaves that are from pairwise distinct species.*

The latter result can be used to show the existence of a reconciliation map determining the extremal event labeling of a given pair of gene and species tree, where the gene tree is least resolved w.r.t. some RBMG.

Lemma 6.6. *Let (T, σ) be some least resolved tree (w.r.t. some RBMG) with extremal event map $\hat{t}_T$ and let $S(W, F)$ be a species tree with $L(S) = \sigma(L(T))$. Then there is a reconciliation $\mu : V(T) \to V(S) \cup E(S)$ that determines the extremal event labeling $\hat{t}_T$.*

Proof. By Cor. 6.2, every inner vertex u with $\hat{t}_T(u) = \bullet$ is only incident to leaves from pairwise distinct species. However, this implies that the set of informative species triples $\mathcal{S}(T, \hat{t}_T, \sigma)$ is empty and thus consistent. Hence, Proposition 6.1 implies that there is a reconciliation map μ from $(T, \hat{t}_T, \sigma)$ to

any species tree S, defined by $\mu(0_T) = 0_S$, $\mu(v) = 0_S\rho_S$ for every inner vertex $v \in V^0(T)$ that is incident to another inner vertex in T, and $\mu(v) = x = \mathrm{lca}_S(\sigma(L(T(v))))$ for any inner vertex v that is only incident to leaves that are from pairwise distinct species, and $\mu(v) = \sigma(v)$ for all leaves of T. By construction of μ, we have $\hat{t}_T(u) = t_T(u)$ with $t_T(u)$ specified by Def. 6.2 for all $u \in V(T)$. □

Suppose that we are given the orthology relation $\Theta(T, \hat{t}_T)$ that is obtained from a least resolved tree (T, σ) explaining some RBMG (G, σ). By Cor. 6.2, every inner vertex u with $\hat{t}_T(u) = \bullet$ is only incident to leaves from pairwise distinct species. Hence, $\Theta(T, \hat{t}_T)$ is the disjoint union of complete graphs. Lemma 6.6 implies that there is always a reconciliation map μ from (T, σ) to any species tree S with $L(S) = \sigma(L(T))$ such that $\hat{t}_T$ is determined by μ as in Def. 6.2. Now we can apply Thm. 6.1 to conclude that all orthologous pairs in $\Theta(T, \hat{t}_T)$ are reciprocal best matches. In other words, all complete graphs of $\Theta(T, \hat{t}_T)$ are also induced subgraphs of the underlying RBMG (G, σ). Hence, $\Theta(T, \hat{t}_T)$ is obtained from (G, σ) by removing edges such that the resulting graph is the disjoint union of cliques, see the top-right tree in Fig. 32 for an example. However, Fig. 32 also shows that many edges have to be removed to obtain $\Theta(T, \hat{t}_T)$. Note that this observation in essence establishes the precise relationship of orthology detection and clustering since (graph) clustering can be interpreted as the graph editing problem for disjoint unions of complete graphs [22].

The results above show that the RBMG contains the orthology relation. Equivalently, RBMGs imply constraints on the event labeling. We also observe that the RBMGs cannot provide conclusive evidence regarding edges that *must* correspond to orthologous pairs. In the following sections we consider the constraints implied by the detailed structure of RBMGs and BMGs in more detail.

6.3 CLASSIFICATION OF RBMGS

This section establishes the connection between hc-cographs, discriminating *hc*-cotrees, and the orthology relation.

Let $\mathcal{C}(G, \sigma)$ be the set of the connected components of the *3-colored* induced subgraphs of an RBMG (G, σ). We have already seen in the last chapter that any $(C, \sigma) \in \mathcal{C}(G, \sigma)$ is either of Type (A), (B), or (C), and the graphs for which all $(C, \sigma) \in \mathcal{C}(G, \sigma)$ are of Type (A) are exactly the RBMGs that are cographs. Intuitively, these co-RBMGs have a close connection to orthology graphs because orthology graphs are cographs. The components of Type (B) and Type (C), in contrast, contain induced P_4s and thus, are not cographs. Thus, by Obs. 6.1, they introduce false positives relative to the orthology graph Θ. We distinguish here co-RBMGs, (B)-*RBMGs*, and (C)-*RBMGs* depending on whether $\mathcal{C}(G, \sigma)$ contains only Type (A) components, at least one Type (B) but not Type (C) component, or at least one Type (C) component.

Recall from the last chapter that co-RBMGs have a convenient structure that can be readily understood in terms of *hc*-cographs (cf. Def. 5.18). Moreover, the recursive construction of (G, σ) also defines a corresponding *hc*-

cotree $(T^G_{hc}, t_{hc}, \sigma)$. According to Thm. 5.10 every co-RBMG (G, σ) is explained by its *hc*-cotree. Let $\{v', v''\} = \mathsf{child}(u)$. If $t_{hc}(u) = 1$, then $\sigma(L(T^G_{hc}(v'))) \cap \sigma(L(T^G_{hc}(v''))) = \emptyset$ in agreement with Lemma 6.2. On the other hand, if $t_{hc}(u) = 0$, then (K3) implies $\sigma(L(T^G_{hc}(v'))) \cap \sigma(L(T^G_{hc}(v''))) \neq \emptyset$, in which case u indeed must be a duplication.

As we have seen in the previous chapter, the cotree $(T^G_{hc}, t_{hc}, \sigma)$ will in general not be discriminating. However, it is not necessarily possible to reduce $(T^G_{hc}, t_{hc}, \sigma)$ to a discriminating *hc*-cotree $(\hat{T}^G_{hc}, \hat{t}_{hc}, \sigma)$ that still explains (G, σ). It is of interest, therefore, to ask whether there are true orthology relations Θ that are not *hc*-cographs, or equivalently, when does a discriminating *hc*-cotree $(\hat{T}^G_{hc}, \hat{t}_{hc}, \sigma)$ that is obtained by edge contraction from a given *hc*-cotree $(T^G_{hc}, t_{hc}, \sigma)$ still explain an RBMG (G, σ)?

Consider an *hc*-cotree $(T^G_{hc}, t_{hc}, \sigma)$ explaining a co-RBMG (G, σ). Since G is a cograph, it also represents an orthology relation, which in turn is represented by a unique discriminating cotree $(\hat{T}^G_{hc}, \hat{t}_{hc}, \sigma)$ that is obtained by contracting all edges uv in T^G_{hc} with $t_{hc}(u) = t_{hc}(v)$ [97]. For every vertex $\hat{u}$ in $\hat{T}^G_{hc}$ with $\hat{t}_{hc}(\hat{u}) = 1$ we still have disjoint color sets $\sigma(L(\hat{T}^G_{hc}(\hat{v}'))) \cap \sigma(L(\hat{T}^G_{hc}(\hat{v}''))) = \emptyset$ for any two children $\hat{v}', \hat{v}'' \in \mathsf{child}(\hat{u})$. Hence, in the absence of HGT (and probably other types of evolutionary events such as recombination, hybridization, or incomplete lineage sorting), the latter property is always satisfied for any true orthology relation.
Condition (K3) implies that the discriminating tree $(\hat{T}^G_{hc}, \hat{t}_{hc}, \sigma)$ still explains (G, σ) if for every $\hat{u}$ with $\hat{t}_{hc}(\hat{u}) = 0$ we have $\sigma(L(\hat{T}^G_{hc}(\hat{v}'))) \subseteq \sigma(L(\hat{T}^G_{hc}(\hat{v}'')))$ or $\sigma(L(\hat{T}^G_{hc}(\hat{v}''))) \subseteq \sigma(L(\hat{T}^G_{hc}(\hat{v}')))$ for all $\hat{v}', \hat{v}'' \in \mathsf{child}(\hat{u})$. Now suppose there are no losses. Then every duplication event has the property that $\sigma(L(T^G_{hc}(v'))) = \sigma(L(T^G_{hc}(v')))$ for all $v', v'' \in \mathsf{child}(u)$ and $t_{hc}(u) = 0$. Clearly, this is still true after contracting all 0-0 edges, i.e., $\sigma(L(\hat{T}^G_{hc}(\hat{v}'))) = \sigma(L(\hat{T}^G_{hc}(\hat{v}')))$ for all $\hat{v}', \hat{v}'' \in \mathsf{child}(\hat{u})$ and $\hat{t}_{hc}(\hat{u}) = 0$. Therefore $(\hat{T}^G_{hc}, \hat{t}_{hc}, \sigma)$ explains (G, σ).
As a consequence we have

Observation 6.2. *In the absence of losses and HGT, $G(T, \sigma)$ is a co-RBMG if and only if $G(T, \sigma)$ is an orthology relation.*

Based on the latter arguments, in the absence of HGT, a true orthology relation cannot be an *hc*-cograph if and only if Condition (K3) is violated for some vertex $\hat{u}$ with $\hat{t}_{hc}(\hat{u}) = 0$ for any of its cotrees and thus, in particular, for its discriminating cotree $(\hat{T}^G_{hc}, \hat{t}_{hc}, \sigma)$. In this case, there are two children $v_1, v_2 \in \mathsf{child}(\hat{u})$ such that $\sigma(L(\hat{T}^G_{hc}(v_1)))$ and $\sigma(L(\hat{T}^G_{hc}(v_2)))$ are not contained in each other. In both cases, there must have been losses in the subsequent history of both v_1 and v_2 such that there are genes $x_1 \neq x_2$ from some species $\sigma(x_1) \neq \sigma(x_2)$ such that $x_i \preceq_{\hat{T}^G_{hc}} v_i$ but $x' \not\preceq_{\hat{T}^G_{hc}} v_j$ for all x' with $\sigma(x') = \sigma(x_i)$, $i, j \in \{1, 2\}$ being distinct. We say that losses leading to the latter case are *non*-hc-*preserving*.

Observation 6.3. *In the absence of HGT and non*-hc-*preserving losses, $G(T, \sigma)$ is a co-RBMG if and only if $G(T, \sigma)$ is an orthology relation.*

Just like for event-labeled trees in general, it is not necessarily possible to reconcile a (discriminating) *hc*-cotree with any species tree. A counterexample

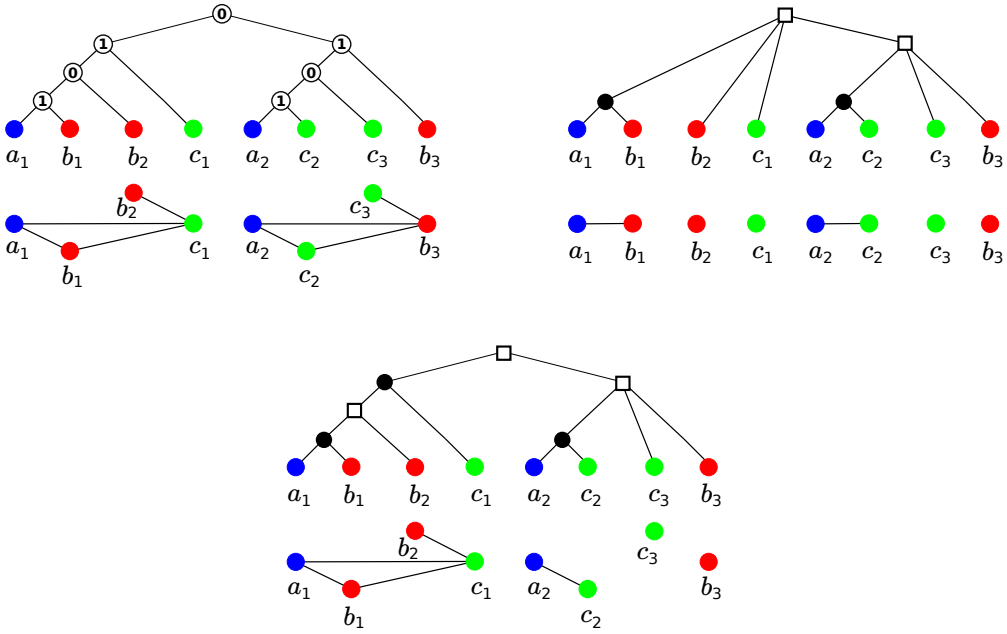


Fig. 32. *Top Left:* A (discriminating) *hc*-cotree $(T^G_{hc}, t_{hc}, \sigma)$. Its corresponding *hc*-cograph $(G, \sigma) = (\Theta(T^G_{hc}, t_{hc}), \sigma)$ is drawn below $(T^G_{hc}, t_{hc}, \sigma)$. *Top Right:* A tree $(T^*, \hat{t}_T, \sigma)$ that is least resolved w.r.t. (G, σ) together with extremal labeling $\hat{t}_T$ and the resulting orthology relation $\Theta(T^*, \hat{t}_T)$, where $(T^*, \hat{t}_T)$ is not discriminating. *Below:* A tree $(T, \hat{t}_T, \sigma)$ together with extremal labeling $\hat{t}_T$ that explains (G, σ) but is not least resolved w.r.t. (G, σ). The resulting orthology relation $\Theta(T, \hat{t}_T)$ is drawn below $(T, \hat{t}_T, \sigma)$.

is shown in Fig. 32. To be more precise, the *hc*-cotree $(T^G_{hc}, t_{hc}, \sigma)$ in Fig. 32 (top left) yields the conflicting species triples $AB|C$ and $AC|B$. Hence, Prop. 6.1 implies that $(T^G_{hc}, t_{hc}, \sigma)$ cannot be reconciled with any species tree although (T^G_{hc}, σ) explains the RBMG (G, σ). One can contract edges of (T^G_{hc}, σ) to obtain a least resolved tree (T^*, σ) that still explains (G, σ), see Fig. 32 (top right). In agreement with Lemma 6.6, $\mathcal{S}(T^*, \hat{t}_T, \sigma) = \emptyset$ and thus, there is always a reconciliation map μ from $(T^*, \hat{t}_T, \sigma)$ to any species tree S with $L(S) = \sigma(L(T))$ such that $\hat{t}_T$ is determined by μ as in Def. 6.2. Moreover, in agreement with Theorem 6.1, all orthologous pairs in $\Theta(T^*, \hat{t}_T, \sigma)$ are best matches. Although (T^*, σ) explains (G, σ), the two graphs $(G, \sigma) = (\Theta(T^G_{hc}, t), \sigma)$ and $(\Theta(T^*, \hat{t}_T), \sigma)$ are very different. In particular, $\Theta(T^*, \hat{t}_T)$ is the disjoint union of cliques.

This observation essentially establishes the precise relationship of orthology detection and clustering techniques as used by many orthology inference tools. However, there is no need to edit (G, σ) to the disjoint union of cliques. An example is provided by the tree $(T, \hat{t}_T, \sigma)$ in Fig. 32 (bottom). Obviously, $\Theta(T, \hat{t}_T)$ is not the disjoint union of cliques. Moreover, there is only one species triple $AB|C$ provided by $(T, \hat{t}_T, \sigma)$. Prop. 6.1 implies that $(T, \hat{t}_T, \sigma)$ can be reconciled with any species tree that displays $AB|C$. In other words, $\Theta(T, \hat{t}_T)$ is already "biologically feasible" and there is no need to remove further edges from $\Theta(T, \hat{t}_T)$.

The aim of this section is a first step to extract the orthology relation Θ from the (reciprocal) best match graph by identifying false positive edges in the latter. Since the orthology relation must be a cograph, it is natural to consider the smallest obstructions, i.e., induced P_4s in more detail.

Recall from Section 5.6 that every induced P_4 in an RBMG contains either three or four distinct colors. We already showed that, with respect to a fixed BMG, every induced P_4 is either a good, a bad, or an ugly quartet (cf. Def. 5.14). Moreover, recall that good, bad, and ugly quartets cannot appear in RBMGs of Type (A) since these are cographs and thus, by definition, do not contain induced P_4s.

The location of good quartets (in contrast to bad and ugly quartets) turns out to be strictly constrained (cf. Lemma 5.35). This fact can be used to show that the "middle" edge of any good quartet must be a false positive orthology assignment. More precisely, we have

Lemma 6.7. *Let T and S be planted trees, $\sigma : L(T) \to L(S)$ a surjective map, and μ a reconciliation map from (T, σ) to S determining an event labeling t_T on T. If $\langle xyzx' \rangle$ is a good quartet in the BMG $\vec{G}(T, \sigma)$, then $t_T(v) = \square$ for $v := \mathrm{lca}(x, x', y, z)$.*

Proof. Lemma 5.35 implies that for a good quartet $\langle xyzx' \rangle$ in $\vec{G}(T, \sigma)$ with $v := \mathrm{lca}(x, x', y, z)$ there are two distinct children $v_1, v_2 \in \mathsf{child}(v)$ such that $x, y \preceq_T v_1$ and $x', z \preceq_T v_2$. Thus, in particular, v_1 and v_2 must be inner vertices in (T, σ). Since $\sigma(x) = \sigma(x')$ by definition of a good quartet, we have $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) \neq \emptyset$. Hence, by Lemma 6.2, $\mu(v) \notin V^0(S)$ which implies $t_T(v) \neq \bullet$. Since v is an inner vertex of T, we can conclude $t_T(v) = \square$. □

As an immediate consequence, this is in particular true for extremal event labelings:

Corollary 6.3. *Let (T, σ) be some leaf-labeled tree and $\hat{t}_T$ the extremal event labeling for (T, σ). If $\langle xyzx' \rangle$ is a good quartet in the BMG $\vec{G}(T, \sigma)$, then $\hat{t}_T(v) = \square$ for $v := \mathrm{lca}(x, x', y, z)$.*

Given an RBMG (G, σ) that contains a good quartet $\langle xyzx' \rangle$ (w.r.t. the underlying BMG $(\vec{G}, \sigma)$), the edge yz therefore always corresponds to a false positive orthology assignment, i.e., it is not contained in the true orthology relation Θ. However, not all false positives can be identified in this way from good quartets. The RBMG $G(T_1, \sigma)$ in Fig. 33, for instance, contains only one good quartet, that is $\langle a_1c_2b_2a_2 \rangle$. After removal of the false positive edge c_2b_2, the remaining undirected graph still contains the bad quartet $\langle a_1b_1c_1a_2 \rangle$, hence, in particular, it still contains an induced P_4 and is, therefore, not an orthology relation.

Neither bad nor ugly quartets can be used to unambiguously identify false positive edges. For an example, consider Fig. 33. The two 3-RBMGs $G(T_1, \sigma)$ and $G(T_2, \sigma)$ both contain the bad quartet $\langle a_1b_1c_1a_2 \rangle$. As a consequence of

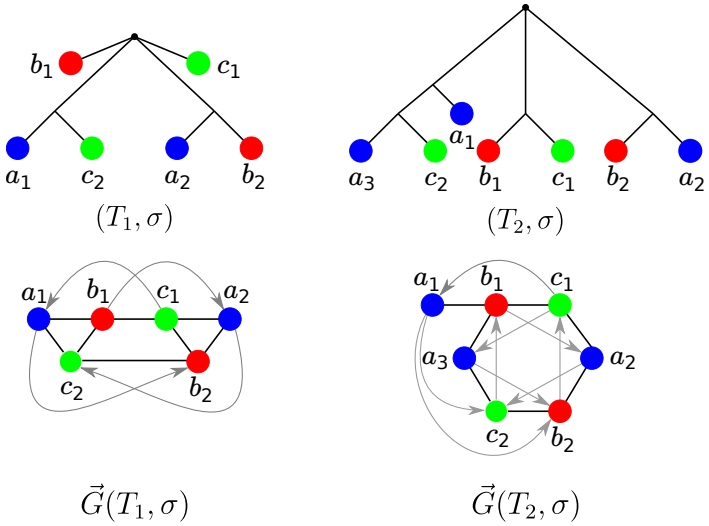


Fig. 33. Not all false positive orthology assignments can be identified using good quartets. Conversely, bad and ugly quartets do not unambiguously identify false positive edges. See the text below Cor. 6.3 for more explanation.

Lemma 6.2, neither the root of T_1 nor the root of T_2 can be labeled by a speciation event. Hence, as a_1, b_1, c_1, a_2 reside all in different subtrees below the root of T_1, all edges a_1b_1, b_1c_1, c_1a_2 in $G(T_1, \sigma)$ correspond to false positive orthology assignments. On the other hand, the vertices b_1 and c_1 reside within the same 2-colored subtree below the root of T_2 and are incident to the same parent in T_2. Therefore one easily checks that there exist reconciliation scenarios where b_1 and c_1 are orthologous, hence the edge b_1c_1 must indeed be contained in the orthology relation. Similarly, $\langle a_1b_1c_1b_2\rangle$ and $\langle a_1b_1a_3c_2\rangle$ are ugly quartets in $G(T_1, \sigma)$ and $G(T_2, \sigma)$, respectively. By the same argumentation as before, the edges a_1b_1, b_1c_1, and c_1b_2 are false positives in $G(T_1, \sigma)$. For (T_2, σ), however, there exist reconciliation scenarios, where a_3 and c_2 are orthologs.

Cor. 5.12, finally, implies that every (B)-RBMG and every (C)-RBMG contains at least one good quartet. In particular, therefore, there is at least one false positive orthology assignment that can be identified with the help of good quartets. We shall see below using simulated data that in practice the overwhelming majority of false positive orthology assignments is already identified by good quartets.

From a theoretical point of view it is interesting nevertheless that it is possible to identify even more false positive orthology assignments starting from Lemma 6.2. We close this section with a brief discussion of this issue. To this end, we extend the leaf sets L^P_*, L^P_s, L^P_t and $L^H_*, L^H_r, L^H_s, L^H_t$ that have been introduced in the Subsections 5.5.4 and 5.5.5 for S-thin 3-RBMGs, to general 3-RBMGs:

Definition 6.5. *Let (G,σ) be a 3-RBMG with vertex set L and colors $S = \{r,s,t\}$, and let $(G/\mathsf{S},\sigma_{/\mathsf{S}})$ with vertex set $\overline{L}$ be its S-thin version. We set*

$$L_s^P := \{x \mid x \in L, [x] \in \overline{L}_s^P\}$$
$$L_t^P := \{x \mid x \in L, [x] \in \overline{L}_t^P\}$$
$$L_*^P := \{x \mid x \in L, [x] \in \overline{L}_*^P\}$$

if (G,σ) is of Type (B) *and $(G/\mathsf{S},\sigma_{/\mathsf{S}})$ B-like w.r.t. some induced path P, and we set*

$$L_r^H := \{x \mid x \in L, [x] \in \overline{L}_r^H\}$$
$$L_s^H := \{x \mid x \in L, [x] \in \overline{L}_s^H\}$$
$$L_t^H := \{x \mid x \in L, [x] \in \overline{L}_t^H\}$$
$$L_*^H := \{x \mid x \in L, [x] \in \overline{L}_*^H\}$$

if (G,σ) is of Type (C) *and $(G/\mathsf{S},\sigma_{/\mathsf{S}})$ C-like w.r.t. some hexagon H.*

The cases of Type (B) and (C) 3-RBMGs will be treated separately, starting with Type (B). We first need a technical result:

Lemma 6.8. *Let (G,σ) be a connected 3-RBMG of Type* (B) *with vertex set L, $(G/\mathsf{S},\sigma_{/\mathsf{S}})$ its S-thin version with vertex set $\overline{L}$, and (T,σ) a leaf-labeled tree that explains (G,σ). Moreover, let $P := \langle[\tilde{x}_1][\tilde{y}][\tilde{z}][\tilde{x}_2]\rangle$ for some good quartet $\langle\tilde{x}_1\tilde{y}\tilde{z}\tilde{x}_2\rangle$ in $\vec{G}(T,\sigma)$, and set $v := \mathrm{lca}_T(\tilde{x}_1,\tilde{x}_2,\tilde{y},\tilde{z})$. Then the leaf sets L_s^P, L_t^P, and L_*^P, where $\sigma(\tilde{x}_1) = \sigma(\tilde{x}_2) = r$, $\sigma(\tilde{y}) = s$, and $\sigma(\tilde{z}) = t$, satisfy:*

(i) $L_t^P, L_s^P \subseteq L(T(v))$,

(ii) *If $L_c^P \cap L(T(v')) \neq \emptyset$ for some $v' \in \mathsf{child}(v)$ and $c \in \{s,t\}$, then*
  (a) $L_{\bar{c}}^P \cap L(T(v')) = \emptyset$, *where $\bar{c} \in \{s,t\}, \bar{c} \neq c$,*
  (b) $\sigma(L(T(v'))) \subseteq \sigma(L_c^P)$,

(iii) $\mathrm{lca}_T(a,b) = v$ *for any $a \in L_*^P$, $b \notin L_*^P$ with $ab \in E(G)$.*

Proof. Throughout this proof we will often use the fact that $xy \in E(G)$ if and only if $[x][y] \in E(G/\mathsf{S})$ for any $x,y \in L$ (cf. Lemma 5.4).

Lemma 5.24 implies $[\tilde{x}_1],[\tilde{y}] \in \overline{L}_t^P$ and $[\tilde{x}_2],[\tilde{z}] \in \overline{L}_s^P$, thus, by definition, we have $\tilde{x}_1,\tilde{y} \in L_t^P$ and $\tilde{x}_2,\tilde{z} \in L_s^P$. Moreover, by Lemma 5.35, there exist distinct children $v_1,v_2 \in \mathsf{child}(v)$ such that $\tilde{x}_1,\tilde{y} \preceq_T v_1$ and $\tilde{x}_2,\tilde{z} \preceq_T v_2$. Therefore $\tilde{y}\tilde{z} \in E(G)$ implies $\sigma(L(T(v_1))) = \{r,s\}$; otherwise there exists a leaf $z' \in L(T(v_1)) \cap L[t]$ which implies $\mathrm{lca}_T(\tilde{y},z') \prec_T v = \mathrm{lca}_T(\tilde{y},\tilde{z})$; a contradiction to $\tilde{y}\tilde{z} \in E(G)$. Analogously we obtain $\sigma(L(T(v_2))) = \{r,t\}$.

(i) By symmetry, it suffices to consider L_t^P in more detail, analogous arguments can then be applied to L_s^P. Let $a \in L_t^P$, or equivalently $[a] \in \overline{L}_t^P$ by definition, and suppose first $\sigma(a) = s$. Then Property (B3.b) implies $[a][\tilde{z}] \in E(G/\mathsf{S})$. As a consequence of Lemma 5.4 we thus have $a\tilde{z} \in E(G)$. Hence, $\tilde{y}\tilde{z} \in E(G)$ implies $\mathrm{lca}_T(a,\tilde{z}) = \mathrm{lca}_T(\tilde{y},\tilde{z}) = v$ and thus, $a \preceq_T v$. We therefore conclude $L_t^P \cap L[s] \subseteq L(T(v))$. Now assume $\sigma(a) = r$. By Property (B2.b), we either

have $N_s([a]) = \emptyset$ or there exists a vertex $y \in L[s]$ such that $[y] \in \overline{L}_t^P$ and $N_s([a]) = \{[y]\}$. In the latter case, since $[y] \in \overline{L}_t^P$ implies $y \in L_t^P$ and, in addition, it holds $L_t^P \cap L[s] \subseteq L(T(v))$, we have $y \preceq_T v$. Moreover, by (B3.b), it holds $[\tilde{x}_2][y] \notin E(G/\mathsf{S})$, hence $\tilde{x}_2 y \notin E(G)$. As a consequence of the latter and the fact that $[a][y] \in E(G/\mathsf{S})$ implies $ay \in E(G)$, it must hold $\mathrm{lca}_T(a, y) \prec_T \mathrm{lca}_T(\tilde{x}_2, y) \preceq_T v$ and thus, $a \preceq_T v$. Otherwise, if $N_s([a]) = \emptyset$, then there must exist a leaf $z \in L[t]$ such that $[z] \in N_t([a])$ due to the connectedness of G/S, which is implied by the connectedness of G (cf. Lemma 5.4). Since $[a] \in \overline{L}_t^P$, Properties (B4.a) and (B4.b) immediately imply $[z] \in \overline{L}_*^P$. Then, by (B4.a), the edge $[\tilde{x}_1][z]$ must be contained in G/S, thus $\tilde{x}_1 z \in E(G)$. Since $\tilde{x}_1, \tilde{z} \preceq_T v$ by Lemma 5.35, it must thus hold $\mathrm{lca}_T(\tilde{x}_1, z) \preceq_T \mathrm{lca}_T(\tilde{x}_1, \tilde{z}) \preceq_T v$. Therefore $\tilde{x}_1 z, az \in E(G)$ implies $\mathrm{lca}_T(a, z) = \mathrm{lca}_T(\tilde{x}_1, z) \preceq_T v$ and thus, $a \preceq_T v$. Hence, $L_t^P \cap L[r] \subseteq L(T(v))$, which finally implies $L_t^P \subseteq L(T(v))$.

(ii) By symmetry, it again suffices to consider the case $c = t$. Let $a \in L_t^P \cap L(T(v'))$ for some $v' \in \mathsf{child}(v)$. Note that, by (i), such a leaf a and inner vertex v' must exist. We need to distinguish the two Cases (1) $\sigma(a) = s$ and (2) $\sigma(a) = r$.

Consider first Case (1), thus in particular $s \in \sigma(L(T(v')))$. Then, as $\sigma(L(T(v_2))) = \{r, t\}$, we have $v' \neq v_2$ and thus, $\mathrm{lca}_T(a, \tilde{z}) = v$. Hence, as $[a][\tilde{z}] \in E(G/\mathsf{S})$ by Property (B3.b) and therefore, $a\tilde{z} \in E(G)$, we can conclude $t \notin \sigma(L(T(v')))$ by analogous arguments as just used for showing $\sigma(L(T(v_1))) = \{r, s\}$. This implies (ii.b). Now assume, for contradiction, that there exists a leaf $x \in L(T(v')) \cap L_s^P$. Since $t \notin \sigma(L(T(v')))$ and, by definition, $s \notin \sigma(L_s^P)$, this leaf x must be of color r. Clearly, either there exists a leaf $y \in L[s]$ such that $xy \in E(G)$ or $N_s(x) = \emptyset$. In the first case, we have $[x][y] \in E(G/\mathsf{S})$ and thus, by (B2.c), $[y] \in \overline{L}_*^P$ which implies $y \in L_*^P$. In particular, as $s \in \sigma(L(T(v')))$ and $xy \in E(G)$ implies $\mathrm{lca}_T(x, y) \preceq_T \mathrm{lca}_T(x, y')$ for any $y' \in L[s]$, we can conclude $y \preceq_T v'$. Moreover, since $[\tilde{x}_2] \in \overline{L}_s^P$, Property (B3.a) implies $[\tilde{x}_2][y] \in E(G/\mathsf{S})$ and thus, $\tilde{x}_2 y \in E(G)$. However, since $v' \neq v_2$, we have $\mathrm{lca}_T(x, y) \preceq_T v' \prec_T v = \mathrm{lca}_T(\tilde{x}_2, y)$; a contradiction to $\tilde{x}_2 y \in E(G)$. We thus conclude $N_s(x) = \emptyset$. Hence, as G is connected, there must exist a leaf z' of color t such that $xz' \in E(G)$, which implies $[x][z'] \in E(G/\mathsf{S})$. By Property (B2.c), we have $[z'] \in \overline{L}_s^P$ and therefore, (B4.b) implies $N_r([z']) = \{[x]\}$. Since $t \notin \sigma(L(T(v')))$, there is a $v'' \in \mathsf{child}(v) \setminus \{v'\}$ such that $z' \preceq_T v'' \prec_T v$. From $xz' \in E(G)$ and $\mathrm{lca}_T(x, z') = v$, we conclude that $r \notin \sigma(L(T(v'')))$. Moreover, Lemma 5.9 implies that there exist leaves $x', y' \in L(T(v'))$ with $\sigma(x') = r$ and $\sigma(y') = s$ such that $x'y' \in E(G)$. Thus, as by assumption $N_s(x) = \emptyset$, we in particular have $[x] \neq [x']$. Since $r \notin \sigma(L(T(v'')))$ and $t \notin \sigma(L(T(v')))$, it follows $x'z' \in E(G)$ and therefore, $[x'] \in N_r([z'])$; a contradiction to $N_r([z']) = \{[x]\}$. This implies (ii.a).

Now consider Case (2), i.e., $\sigma(a) = r$. We first show that $\sigma(L(T(v'))) \subsetneq \{r, s, t\}$ holds. Assume, for contradiction, that $L(T(v'))$ contains leaves $y \in L[s]$ and $z \in L[t]$. If $v' \neq v_2$, this implies $\mathrm{lca}_T(y, z) \prec_T v = \mathrm{lca}(y, \tilde{z})$ and thus, $y\tilde{z} \notin E(G)$ and in particular $[y][\tilde{z}] \notin E(G/\mathsf{S})$; a contradiction to (B4.b). One analogously obtains a contradiction for the case $v' \neq v_1$; therefore $\sigma(L(T(v'))) \subsetneq \{r, s, t\}$ and we either have $\sigma(L(T(v'))) \subseteq \{r, s\}$ or $\sigma(L(T(v'))) \subseteq \{r, t\}$. If

$\sigma(L(T(v'))) = \{r\}$, then it clearly holds $N(x) = N(a)$ and thus $x \in L_t^P$ for any $x \in L(T(v'))$, hence (ii.a) and (ii.b) are trivially satisfied. If $\sigma(L(T(v'))) = \{r, s\}$, then (ii.b) is trivially satisfied. Moreover, by Lemma 5.9, $L(T(v'))$ contains leaves $x' \in L[r]$ and $y' \in L[s]$ such that $x'y' \in E(G)$. Hence, we have $[x'][y'] \in E(G/\mathsf{S})$ and Property (B4.b) implies $[y'][\tilde{z}] \in E(G/\mathsf{S})$ and thus, $y'\tilde{z} \in E(G)$. As $\sigma(L(T(v_2))) = \{r, t\}$ and $\sigma(L(T(v'))) = \{r, s\}$, we clearly have $v' \neq v_2$ and thus, $\mathrm{lca}_T(x', y') \preceq_T v' \prec_T v = \mathrm{lca}_T(\tilde{x}_2, y')$. Hence, $\tilde{x}_2 y' \notin E(G)$, which implies $N([y']) \neq \overline{L}_s^P \cup (\overline{L}_*^P \setminus \{[y']\})$ since $\tilde{x}_2 \in L_s^P$. Therefore, by Property (B3.a), we have $[y'] \notin \overline{L}_*^P$, implying $y' \notin L_*^P$. We thus conclude $y' \in L_t^P$. Hence, we can apply the argumentation of Case (1) (by substituting $a = y'$) in order to infer (ii.a).
Finally, for contradiction, assume $\sigma(L(T(v'))) = \{r, t\}$. In particular, this implies $v_1 \neq v'$. Clearly, either there exists a leaf $y \in L[s]$ such that $ay \in E(G)$ (and thus $[a][y] \in E(G/\mathsf{S})$) or $N_s(a) = \emptyset$. In the latter case, since G is connected, there must be a leaf $z \in L[t]$ such that $az \in E(G)$ and $[a][z] \in E(G/\mathsf{S})$. In particular, as $\sigma(L(T(v'))) = \{r, t\}$, this implies $z \preceq_T v'$. By (B2.b), we have $[z] \in \overline{L}_*^P$ and thus, by (B4.a), it follows $[\tilde{x}_1][z] \in E(G/\mathsf{S})$ implying $\tilde{x}_1 z \in E(G)$; a contradiction since $\mathrm{lca}_T(z, a) \preceq_T v' \prec_T v = \mathrm{lca}_T(z, \tilde{x}_1)$. Hence, there must exist a leaf $y \in L[s]$ such that $ay \in E(G)$. By (B2.b), we have $N_s([a]) = \{[y]\}$ and $[y] \in \overline{L}_t^P$. Then (B3.b) implies $N_r([y]) \subset \overline{L}_t^P$. It is easy to see that this implies $N_r(y) \subset L_t^P$. Since $s \notin \sigma(L(T(v')))$, there must exist a vertex $v'' \in \mathsf{child}(v) \setminus \{v'\}$ such that $y \preceq_T v'' \prec_T v = \mathrm{lca}_T(a, y)$. One easily checks that $ay \in E(G)$ implies $r \notin \sigma(L(T(v'')))$. Together with $\sigma(L(T(v_2))) = \{r, t\}$, this implies $\mathrm{lca}_T(\tilde{x}_2, y) = v \preceq_T \mathrm{lca}_T(x'', y)$ and $\mathrm{lca}_T(\tilde{x}_2, y) = v \preceq_T \mathrm{lca}_T(\tilde{x}_2, y')$ for any $x'' \in L[r]$ and $y' \in L[s]$. Thus, $\tilde{x}_2 y \in E(G)$, which, as $\tilde{x}_2 \in L_s^P$, contradicts $N_r(y) \subset L_t^P$. We therefore conclude that $\sigma(L(T(v'))) = \{r, t\}$ is not possible, which finally completes the proof.

(iii) Since, by definition, $V(G)$ is partitioned into L_s^P, L_t^P, and L_*^P, the leaf b must be either contained in L_t^P or L_s^P. Suppose first $b \in L_t^P$. Since $[a][b] \in E(G/\mathsf{S})$ follows from $ab \in E(G)$, Properties (B2.a), (B3.a), and (B4.a) immediately imply $\sigma(a) = t$. Moreover, by (i), there exists some $v' \in \mathsf{child}(v)$ such that $b \preceq_T v' \prec_T v$, and, by (ii.b), $\sigma(L(T(v'))) \subseteq \sigma(L_t^P) = \{r, s\}$. Hence, as $\sigma(a) = t$, we can conclude $\mathrm{lca}_T(a, b) \succeq_T v$. Similarly, $\sigma(L(T(v'))) \subseteq \{r, s\}$ implies $\mathrm{lca}_T(b, \tilde{z}) = v$, thus it must hold $\mathrm{lca}_T(a, b) \preceq_T \mathrm{lca}_T(b, \tilde{z}) = v$ because of $ab \in E(G)$. In summary, this implies $\mathrm{lca}_T(a, b) = v$. Analogous arguments can be applied to the case $b \in L_s^P$. □

Lemma 6.8 can now be used to identify a potentially very large set of edges that cannot be present in the orthology graph Θ.

Theorem 6.2. *Let T and S be planted trees, $\sigma : L(T) \to L(S)$ a surjective map, and μ a reconciliation map from (T, σ) to S determining an event labeling t_T on T. Moreover, let the leaf sets L_t^P, L_s^P, and L_*^P be defined w.r.t. P, which is the S-thin version of some good quartet of the form (r, s, t, r) in $(\vec{G}, \sigma)$ with color set $S = \{r, s, t\}$. Then $t_T(\mathrm{lca}_T(a, b)) = \square$ for any edge $ab \in E(G)$ such that $a \in L_\star^P$ and $b \notin L_\star^P$, where $\star \in \{s, t, *\}$.*

Proof. Let $P = \langle [x_1][y][z][x_2] \rangle$, i.e., in particular $\sigma(x_1) = \sigma(x_2) = r$, $\sigma(y) = s$, and $\sigma(z) = t$, and let $v := \mathrm{lca}_T(x_1, x_2, y, z)$. Then, by Lemma 5.35, there exist distinct $v_1, v_2 \in \mathsf{child}(v)$ such that $x_1, y \preceq_T v_1$ and $x_2, z \preceq_T v_2$. As $[x_1], [y] \in \overline{L}_t^P$ and $[x_2], [z] \in \overline{L}_s^P$ by Lemma 5.24 and thus, by definition, $x_1, y \in L_t^P$ and $x_2, z \in L_s^P$, Lemma 6.8(ii.b) in particular implies $\sigma(L(T(v_1))) = \{r, s\}$ and $\sigma(L(T(v_2))) = \{r, t\}$.
Now, if $a \in L_t^P$, $b \in L_s^P$, it follows from Lemma 6.8(ii.a) that $\mathrm{lca}_T(a, b) = v$. On the other hand, if $a \in L_*^P$ and either $b \in L_s^P$ or $b \in L_t^P$, then we also have $\mathrm{lca}_T(a, b) = v$ by Lemma 6.8(iii). Since $\sigma(L(T(v_1))) \cap \sigma(L(T(v_2))) = \{r\} \neq \emptyset$, we conclude from Lemma 6.2 that $\mu(v) \notin V^0(S)$, which implies $t_T(v) \neq \bullet$. Therefore we have $t_T(v) = \square$. □

A similar procedure will be applied to Type (C) 3-RBMGs, again starting with an analogous technical result:

Lemma 6.9. *Let (G, σ) be a connected 3-RBMG of Type* (C) *with vertex set L, $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$ its S-thin version with vertex set $\overline{L}$, and (T, σ) a leaf-labeled tree that explains (G, σ). Moreover, let $H := \langle [\tilde{x}_1][\tilde{y}_1][\tilde{z}_1][\tilde{x}_2][\tilde{y}_2][\tilde{z}_2] \rangle$ for some induced hexagon $\langle \tilde{x}_1\tilde{y}_1\tilde{z}_1\tilde{x}_2\tilde{y}_2\tilde{z}_2 \rangle$ in $\vec{G}(T, \sigma)$ with $|N_t([\tilde{x}_1])| > 1$ and $\sigma(\tilde{x}_1) = \sigma(\tilde{x}_2) = r$, $\sigma(\tilde{y}_1) = \sigma(\tilde{y}_2) = s$, and $\sigma(\tilde{z}_1) = \sigma(\tilde{z}_2) = t$, and set $v := \mathrm{lca}_T(\tilde{x}_1, \tilde{x}_2, \tilde{y}_1, \tilde{y}_2, \tilde{z}_1, \tilde{z}_2)$. Then the leaf sets L_r^H, L_s^H, L_t^H, and L_*^H satisfy:*

(i) $L_r^H, L_s^H, L_t^H \subseteq L(T(v))$,

(ii) If $L_c^H \cap L(T(v')) \neq \emptyset$ *for some* $v' \in \mathsf{child}(v)$ *and* $c \in \{r, s, t\}$, *then*
  (a) $L_{\bar{c}}^H \cap L(T(v')) = \emptyset$, *where* $\bar{c} \in \{r, s, t\}, \bar{c} \neq c$,
  (b) $\sigma(L(T(v'))) \subseteq \sigma(L_c^H)$,

(iii) $\mathrm{lca}_T(a, b) = v$ *for any* $a \in L_*^H$, $b \notin L_*^H$ *with* $ab \in E(G)$.

Proof. The proof of Lemma 6.9 closely follows the arguments leading to Lemma 6.8. In particular, we again use the fact that $xy \in E(G)$ if and only if $[x][y] \in E(G/\mathsf{S})$ for any $x, y \in L$ (cf. Lemma 5.4).

By Lemma 5.26, we have $[\tilde{x}_1], [\tilde{y}_1] \in \overline{L}_t^H$, $[\tilde{x}_2], [\tilde{z}_1] \in \overline{L}_s^H$, and $[\tilde{y}_2], [\tilde{z}_2] \in \overline{L}_r^H$, hence $\tilde{x}_1, \tilde{y}_1 \in L_t^H$, $\tilde{x}_2, \tilde{z}_1 \in L_s^H$, and $\tilde{y}_2, \tilde{z}_2 \in L_r^H$. Moreover, by Lemma 5.38(iii), there exist distinct children $v_1, v_2, v_3 \in \mathsf{child}(v)$ such that $\tilde{x}_1, \tilde{y}_1 \preceq_T v_1$, $\tilde{x}_2, \tilde{z}_2 \preceq_T v_2$, and $\tilde{y}_2, \tilde{z}_2 \preceq_T v_3$. In particular, since $\tilde{y}_1\tilde{z}_1 \in E(G)$, it must hold $\sigma(L(T(v_1))) = \{r, s\}$ as otherwise there exists a leaf $z' \in L(T(v_1)) \cap L[t]$ which implies $\mathrm{lca}_T(\tilde{y}_1, z') \prec_T v = \mathrm{lca}_T(\tilde{y}_1, \tilde{z}_1)$; a contradiction to $\tilde{y}_1\tilde{z}_1 \in E(G)$. One analogously checks $\sigma(L(T(v_2))) = \{r, t\}$ and $\sigma(L(T(v_3))) = \{s, t\}$.

(i) By symmetry, it suffices to consider L_t^H in more detail, analogous arguments can then be applied to L_s^H and L_r^H. Let $a \in L_t^H$, or equivalently $[a] \in \overline{L}_t^H$, and suppose first $\sigma(a) = r$. Then Property (C2.b) implies $[a][\tilde{z}_2] \in E(G/\mathsf{S})$ and thus, $a\tilde{z}_2 \in E(G)$. As $\tilde{x}_1\tilde{z}_2 \in E(G)$, we thus have $\mathrm{lca}_T(a, \tilde{z}_2) = \mathrm{lca}_T(\tilde{x}_1, \tilde{z}_2) = v$, hence $a \preceq_T v$. We therefore conclude $L_t^H \cap L[r] \subseteq L(T(v))$. Analogously, we obtain $a \preceq_T v$ for $\sigma(a) = s$ as a consequence of Property (C3.b). In summary, we obtain $L_t^H \subseteq L(T(v))$.

(ii) Again invoking symmetry, it suffices to consider the case $c = t$. Let $a \in L_t^H \cap L(T(v'))$ for some $v' \in \mathsf{child}(v)$. First, let $\sigma(a) = r$. Then, as $r \notin \sigma(L(T(v_3)))$,

we have $v' \neq v_3$ and thus, $\mathrm{lca}_T(a, \tilde{z}_2) = v$. Hence, as $[a][\tilde{z}_2] \in E(G/\mathsf{S})$ by Property (C2.b) and thus $a\tilde{z}_2 \in E(G)$, we can conclude $t \notin \sigma(L(T(v')))$ using the same line of reasoning used above for showing $\sigma(L(T(v_1))) = \{r, s\}$. This implies (ii.b). Now assume, for contradiction, that there exists either (1) a leaf $x \in L(T(v')) \cap L_s^H$ or (2) a leaf $y \in L(T(v')) \cap L_r^H$.
In Case (1), since $t \notin \sigma(L(T(v')))$ and, by definition, $s \notin \sigma(L_s^H)$, this leaf x must be of color r. In particular, since L_s^H and L_t^H are disjoint, we have $x \neq a$. Hence, it must hold $s \in \sigma(L(T(v')))$ as otherwise $N(x) = N(a)$; contradicting $a \in L_t^H$, $x \in L_s^H$, and $L_s^H \cap L_t^H = \emptyset$. This immediately implies $v' \neq v_2$ because $s \notin \sigma(L(T(v_2)))$. By Property (C2.c), as $[\tilde{y}_2] \in \overline{L}_r^H[s]$, we have $[x][\tilde{y}_2] \in E(G/\mathsf{S})$ and thus, $x\tilde{y}_2 \in E(G)$. However, since $s \in \sigma(L(T(v')))$, there exists a leaf $y' \preceq_T v'$ with $\sigma(y') = s$, which implies $\mathrm{lca}_T(x, y') \preceq_T v' \prec_T v = \mathrm{lca}_T(x, \tilde{y}_2)$ because of $\tilde{y}_2 \preceq_T v_3 \neq v'$; a contradiction to $x\tilde{y}_2 \in E(G)$.
Hence, assume Case (2), i.e., there exists $y \in L(T(v')) \cap L_r^H$. Since $t \notin \sigma(L(T(v')))$ and, by definition, $r \notin \sigma(L_r^H)$, the leaf y must be of color s, which in particular implies $v' \neq v_2$. As $t \notin \sigma(L(T(v')))$ and $s \notin \sigma(L(T(v_2)))$, one easily checks that $y\tilde{z}_1 \in E(G)$. However, as $y \in L_r^H$ and thus $[y] \in \overline{L}_r^H$, Property (C3.c) implies $[\tilde{z}_1] \in \overline{L}_r^H$, hence $\tilde{z}_1 \in L_r^H$; a contradiction since $\tilde{z}_1 \in L_s^H$.
In summary, we conclude that $L_{\bar{c}}^H \cap L(T(v')) = \emptyset$, where $\bar{c} \in \{r, s\}$, hence (ii.a) is satisfied for $c = t$. Analogous arguments can be used to demonstrate that properties (ii.a) and (ii.b) are also satisfied for $\sigma(a) = s$.

(iii) Since, by definition, $V(G)$ is partitioned into L_r^H, L_s^H, L_t^H, and L_*^H, the leaf b must be either contained in L_r^H, L_s^H, or L_t^H. Suppose first $b \in L_t^H$. Then, since $[a][b] \in E(G/\mathsf{S})$ follows from $ab \in E(G)$, Properties (C2.a), (C3.a), and (C4.a) immediately imply $\sigma(a) = t$. Moreover, by (i), there exists some $v' \in \mathsf{child}(v)$ such that $b \preceq_T v' \prec_T v$ and, by (ii.b), $\sigma(L(T(v'))) \subseteq \sigma(L_t^H) = \{r, s\}$. Hence, as $\sigma(a) = t$, we can conclude $\mathrm{lca}_T(a, b) \succeq_T v$. Similarly, $\sigma(L(T(v'))) \subseteq \{r, s\}$ implies $\mathrm{lca}_T(b, \tilde{z}_1) = v$, thus it must hold $\mathrm{lca}_T(a, b) \preceq_T \mathrm{lca}_T(b, \tilde{z}_1) = v$ because of $ab \in E(G)$. In summary, this implies $\mathrm{lca}_T(a, b) = v$. Analogous arguments can be applied to the cases $b \in L_s^H$ and $b \in L_r^H$. □

Similar to Type (B) 3-RBMGs, we use Lemma 6.9 to finally identify false positive edges.

Theorem 6.3. *Let T and S be planted trees, $\sigma : L(T) \to L(S)$ a surjective map, and μ a reconciliation map from (T, σ) to S determining an event labeling t_T on T. Moreover, let the leaf sets L_r^H, L_s^H, L_t^H, and L_*^H be defined w.r.t. H, which is the S-thin version of some hexagon $H' = \langle x_1y_1z_1x_2y_2z_2\rangle$ of the form (r, s, t, r, s, t) and $|N_t(x_1) > 1|$ in $(\vec{G}, \sigma)$ with color set $S = \{r, s, t\}$. Then $t_T(\mathrm{lca}_T(a, b)) = \square$ for any edge $ab \in E(G)$ such that $a \in L_\star^H$ and $b \notin L_\star^H$, where $\star \in \{r, s, t, *\}$.*

Proof. Let $v := \mathrm{lca}_T(x_1, x_2, y_1, y_2, z_1, z_2)$. Again, we have $[x_1], [y_1] \in \overline{L}_t^H$, $[x_2], [z_1] \in \overline{L}_s^H$, and $[y_2], [z_2] \in \overline{L}_r^H$ by Lemma 5.26 and thus, $x_1, y_1 \in L_t^H$, $x_2, z_1 \in L_s^H$, $y_2, z_2 \in L_r^H$. Moreover, by Lemma 5.38(iii), there exist distinct $v_1, v_2, v_3 \in \mathsf{child}(v)$ such that $x_1, y_1 \preceq_T v_1$, $x_2, z_1 \preceq_T v_2$, and $y_2, z_2 \preceq_T v_3$. As $x_1, y_1 \in L_t^H$, $x_2, z_1 \in L_s^H$, $y_2, z_2 \in L_r^H$, Lemma 6.9(ii.b) in particular implies $\sigma(L(T(v_1))) = \{r, s\}$, $\sigma(L(T(v_2))) = \{r, t\}$, and $\sigma(L(T(v_3))) = \{s, t\}$.

Now, if $a \in L_c^H$, $b \in L_{\bar{c}}^H$, where $c = \{r, s, t\}$ and $\bar{c} \in \{r, s, t\}, \bar{c} \neq c$, it follows from Lemma 6.9(ii.a) that $\mathrm{lca}_T(a, b) = v$. On the other hand, if $a \in L_*^H$ and $b \in L_c^H$, then we also have $\mathrm{lca}_T(a, b) = v$ by Lemma 6.9(iii). Since $\sigma(L(T(v_i))) \cap \sigma(L(T(v_j))) \neq \emptyset$ for $1 \leq i < j \leq 3$, we conclude from Lemma 6.2 that $\mu(v) \notin V^0(S)$, which implies $t_T(v) \neq \bullet$. Therefore we have $t_T(v) = \square$. □

These results show that if x and y are located in two distinct leaf sets L_s^P, L_t^P, L_*^P in a connected component of Type (B) of an induced 3-RBMG, then $t_T(\mathrm{lca}(x, y)) = \square$. Similarly, edges connecting two vertices of the leaf sets L_r^H, L_s^H, L_t^H, L_*^H in a connected component of Type (C) of an induced 3-RBMG cannot belong to orthologs. However, the simulation study in the following section suggests that such cases that are not covered already by good quartets seem to be exceedingly rare and thus, of little practical relevance.

6.5 SIMULATIONS

Although the edges in an RBMG cannot identify orthologous pairs with certainty (as a consequence of Lemma 6.3), there is a close resemblance in practice, i.e., for empirically determined scenarios. In order to explore this connection in more detail, we consider simulated evolutionary scenarios (T, S, μ). These uniquely determine both the (reciprocal) best match graph $\vec{G}(T, \sigma)$ and $G(T, \sigma)$, resp., and the orthology graph Θ, thus allowing a direct comparison of these graphs. Since only scenarios (T, S, μ) will be analyzed here, simulations tools such as `ALF` [42] that are designed to simulate sequence data, are not used in these simulations. Preliminary simulations have been performed in the context of a Bachelor's thesis [151].

6.5.1 *Method*

In order to simulate evolutionary scenarios (T, S, μ), a stepwise procedure is employed:

(1) **Construction of the species tree S.** The tree S is regarded as ultrametric, i.e., its branch lengths are interpreted as real-time. Given a user-defined number of species N, the species tree S is generated under the *innovations model* as described by Keller-Schmidt and Klemm [126]. The binary trees generated by this model have similar depth and imbalances as those of real phylogenetic trees from databases.

(2) **Construction of the true gene tree $\tilde{T}$.** Traversing the species tree S top-down, one gene tree $\tilde{T}$ is generated with user-defined rates r_D for duplications, r_L for gene losses, and r_H for horizontal transfer events. The number of events along each edge of the species tree, of each event type, is drawn from a Poisson distribution with parameter $\lambda = \ell r_e$, where ℓ is the length of the edge e and r_e is the rate of the event type. Duplication and horizontal transfer events duplicate an active lineage and occur only inside edges of S. For duplications, both offspring lineages remain inside the

same edge of the species tree as their parental gene. In contrast, one of the two offspring lineages of an HGT event is transferred to another, randomly selected, branch of the species tree at the same time. At speciation nodes all branches of the gene tree are copied into each offspring. Loss events terminate branches of $\tilde{T}$. In our setting, loss events may occur only within edges of the species tree that harbor more than one branch of the gene tree, which ensures that every leaf of S is reached by at least one branch of the gene tree $\tilde{T}$. All vertices v of $\tilde{T}$ are labeled with their event type $t_T(v)$, in particular, there are different leaf labels for extant genes and lost genes. The simulation explicitly records the reconciliation map, i.e., the assignment of each vertex of $\tilde{T}$ to a vertex or edge of S.

(3) **Construction of the observable gene tree T from $\tilde{T}$.** The leaves of $\tilde{T}$ are either observable extant genes or unobservable losses. As described by Hernandez-Rosales et al. [105], $\tilde{T}$ is pruned in bottom-up order by removing all loss events and omitting all inner vertices with only a single remaining child.

Using Steps (1) and (2), the simulation consists of 10,000 scenarios for species trees with 3 to 100 species (=leaves) and additional 4,000 scenarios for species trees with 3 to 50 leaves, drawn from a uniform distribution. For each of these species trees, exactly one gene tree was simulated as described above. The rate parameters have been varied between 0.65 and 0.99 in steps of 0.01 for duplication and loss events. For HGTs, either a rate of 0 or a rate in the range between 0.1 and 0.25, again in steps of 0.01, was used. A detailed list of all simulated scenarios can be found in Geiß et al. [74, Supplemental Material].
For each of the 14,000 true gene trees $\tilde{T}$ the total number S_n of speciations, L_n of losses, D_n of duplications, and H_n of HGTs was determined. Summary statistics of the simulated scenarios can be found in Geiß et al. [74, Supplemental Material].

From each true gene tree $\tilde{T}$ the observable gene tree T was extracted as described in Step (3). For all retained vertices, the event labeling t_T and the reconciliation map μ remain unchanged. Since $\mathrm{lca}_T(x,y) = \mathrm{lca}_{\tilde{T}}(x,y)$ for all extant genes $x, y \in L(T)$, it suffices to consider T. The leaf coloring map $\sigma : L(T) \to L(S)$ is obtained from its definition, i.e., setting $\sigma(v) = \mu(v)$ for all $v \in L(T)$. The orthology relation and (reciprocal) best match relation can now be extracted from each scenario.

The orthology relation $\Theta(T, t_T)$ is easily constructed from the event-labeled gene tree (T, t_T) by a simple recursive construction. More precisely, for each $v \in \tilde{T}$ we define a graph $\Theta(v)$ recursively: if v is a leaf, then $\Theta(v)$ is the K_1 with vertex set $\{v\}$ whenever v is an extant gene and $\Theta(v) = \emptyset$, i.e., the empty graph, if v is a loss event. For inner vertices we set

$$\Theta(v) = \begin{cases} \bigtriangledown_{u \in \mathsf{child}(v)} \Theta(u) & \text{if } t(v) = \bullet \\ \bigcup_{u \in \mathsf{child}(v)} \Theta(u) & \text{otherwise} \end{cases} \tag{20}$$

Since $H \triangledown \emptyset = H \cup \emptyset = H$, there is no contribution of the leaves corresponding to a loss event. Thus the graph $\Theta(v)$ can be computed in exactly the same manner from the observable gene tree T and the true gene tree $\tilde{T}$. Hence, $\Theta(\rho_T) = \Theta(\rho_{\tilde{T}}) =: \Theta$ is the orthology graph of the scenario. Note that the planted root 0_T does not appear as the last common ancestor of any two leaves in L, hence it suffices to consider the root ρ_T. Although the next result is an immediate consequence of the definition of cographs and their corresponding cotrees [38], we give here an alternative short proof.

Lemma 6.10. *Let (T, t_T, σ) be an event-labeled and leaf-labeled tree. Then $xy \in E(\Theta(v))$ if and only if $t_T(\mathrm{lca}_T(x, y)) = \bullet$.*

Proof. We proceed by induction. The assertion is trivially true if v is a leaf, in which case $x = y = v$ and thus, $t_T(\mathrm{lca}_T(x, y)) = \odot$; indeed, $xy \notin E(\Theta(v))$ since Θ is loop-free by definition. Now suppose the assertion holds for all $u \prec_T v$ and consider two vertices $x, y \in L(T(v))$. We consider two cases: (i) If $\mathrm{lca}_T(x, y) \prec_T v$, then there is a child $u \in \mathsf{child}(v)$ such that $x, y \in L(T(u))$ and thus, $xy \in E(\Theta(v))$ if and only if $xy \in E(\Theta(u))$, which by induction hypothesis is true if and only if $t_T(\mathrm{lca}(x, y)) = \bullet$. (ii) If $\mathrm{lca}_T(x, y) = v$, then there are two distinct children $u_1, u_2 \in \mathsf{child}(v)$ with $x \preceq_T u_1$ and $y \preceq_T u_2$. The definition of the disjoint union and the join of graphs, resp., implies that $xy \in E(\Theta(v))$ if and only the join is used to combine $\Theta(u_1)$ and $\Theta(u_2)$, i.e., if and only if $t_T(\mathrm{lca}(x, y)) = t_T(v) = \bullet$. □

We already argued in Section 6.3 that, in the absence of losses and HGT events, every duplication event u in a cotree $(T^G_{hc}, t_{hc}, \sigma)$ explaining some RBMG (G, σ) satisfies $\sigma(L(T^G_{hc}(v'))) = \sigma(L(T^G_{hc}(v'')))$ for any $v', v'' \in \mathsf{child}(u)$ in the absence of gene loss and HGT. Similarly, if u is a speciation event, then $\sigma(L(T^G_{hc}(v'))) \cap \sigma(L(T^G_{hc}(v''))) = \emptyset$ for any $v', v'' \in \mathsf{child}(u)$. In particular, we discussed that the discriminating cotree $(\hat{T}^G_{hc}, \hat{t}_{hc}, \sigma)$ still satisfies these properties and explains (G, σ). Hence, together with Obs. 6.2, Lemma 6.10 immediately yields

Observation 6.4. *In the absence of losses and HGT, it holds $(G, \sigma) = (\Theta(T^G_{\mathrm{hc}}, t_{\mathrm{hc}}), \sigma)$ for any* hc*-cotree $(T^G_{\mathrm{hc}}, t_{\mathrm{hc}}, \sigma)$ explaining a co-RBMG (G, σ).*

By construction, $\Theta(u)$ is an induced subgraph of $\Theta(v)$ whenever $u \preceq_T v$. It is thus sufficient to store the binary $|L| \times |L|$ adjacency matrix of Θ. Traversing T in postorder, one sets $\Theta_{xy} = 1$, i.e., $xy \in E(\Theta)$, for all xy with $x \in L(T(u_1))$ and $y \in L(T(u_2))$ where u_1 and u_2 are distinct children of v, if and only if v is a speciation vertex. Since the pair x, y is considered exactly once, namely when $v = \mathrm{lca}(x, y)$ is encountered in the traversal of T, the total effort is $O(|L|^2)$.

The computation of the BMG $\vec{G}(T, \sigma)$ proceeds as follows: first every inner vertex v is associated with the lists $L_r(v) := \{x \in L(T(v)) | \sigma(x) = r\}$ of leaves below v with color r. We have $L_r(v) = \bigcup_{u \in \mathsf{child}(v)} L_r(u)$ for inner vertices, while leaves are initialized with $L_r(v) = \{v\}$ if $\sigma(v) = r$, and $L_r(v) = \emptyset$ if $\sigma(v) \neq r$. Again this can be achieved in not more than quadratic time. Now define $C_{\neg s}(v) := \{u \in \mathsf{child}(v) | L_s(u) = \emptyset\}$ and $C_s(v) := \{u \in \mathsf{child}(v) | L_s(u) \neq \emptyset\}$.

Lemma 6.11. *Let u_1 and u_2 be two distinct children of some inner vertex v of the leaf-colored tree (T, σ) and let $x \in L(T(u_1))$ with $\sigma(x) = r$ and $y \in L(T(u_2))$ with $\sigma(y) = s \neq r$. Then (x, y) is a best match in (T, σ) if and only if*

$$u_1 \in C_r(v) \cap C_{\neg s}(v) \quad \text{and} \quad u_2 \in C_s(v).$$

Proof. If $L_s(u_1) = \emptyset$, then there is no best match of color s for x in $L(T(u_1))$, i.e., any best match $\sigma(y') = s$ satisfies $v \preceq \mathrm{lca}(x, y')$. From $\mathrm{lca}(x, y) = v$ we see that (x, y) is indeed a best match. On the other hand, if $L_s(u_1) \neq \emptyset$, then there is a leaf $y' \in L_s(u_1)$ with $\mathrm{lca}(x, y') \preceq u_1 \prec v = \mathrm{lca}(x, y)$ and thus, y is not a best match for x. □

Algorithm 6 Construction of $\vec{G}(T, \sigma)$

Require: Leaf-colored tree (T, σ)
  for all leaves v of T, colors r **do**
    $L(T(v)) = \{v\}$
    if $\sigma(v) = r$ **then**
      $\ell_{vr} = 1$
    else
      $\ell_{vr} = 0$
  for all inner vertices v of T in postorder **do**
    for all $u_1, u_2 \in \mathsf{child}(v)$, $u_1 \neq u_2$ **do**
      for all $x \in L(T(u_1))$ and $y \in L(T(u_2))$ **do**
        $(x, y) \in \vec{G}(T, \sigma)$ **if** $\ell_{u_1, \sigma(y)} = 0$
    $L(T(v)) = \bigcup_{u \in \mathsf{child}(v)} L(T(u))$
    for all $u \in \mathsf{child}(v)$, colors $r \in S$ **do**
      $\ell_{vr} = 1$ if $\ell_{ur} = 1$

This observation yields the very simple way to construct $\vec{G}(T, \sigma)$. Algorithm 6 iterates over all pairs of vertices $x, y \in L$ such that each pair is visited exactly once by considering for every interior vertex v exactly the pairs that are members of two distinct subtrees rooted at children u_1 and u_2 of v. Since $y \in L_{\sigma(y)}(u_2)$ and $x \in L_{\sigma(x)}(u_1)$ is guaranteed by construction, (x, y) is a best match if and only if $L_{\sigma(y)}(u_1) = \emptyset$ by Lemma 6.11. Using the precomputed binary variable ℓ_{vr} with value 1 if $L_r(v) \neq \emptyset$ and $\ell_{vr} = 0$ otherwise, this can be done in constant time $O(|L|)$. By traversing T in postorder, finally, the lists of leaves $L(v)$ can be computed on the fly. Since no subtree is revisited, there is no need to retain the $L(T(u))$ for the children, i.e., for each vertex v, the lists of its children can simply be concatenated. Similarly, the variables ℓ_{vr} can be obtained while traversing T using the fact that $\ell_{vr} = 1$ if and only if $\ell_{ur} = 1$ for at least one of its children. Hence, Algorithm 6 runs in time with $O(|L|\,|S|)$ memory using a single postorder traversal of T.
The RBMG $G(T, \sigma)$ is easily obtained from the BMG $\vec{G}(T, \sigma)$ by extracting its symmetric part. Clearly, the effort for this step is also bounded by $O(|L|^2)$. Note, finally, that given (T, t_T, σ), both the orthology graph Θ and the BMG $G(T, \sigma)$ can be found in $O(|L|^2)$ time using Tarjan's off-line lowest common ancestors algorithm [219, 70] to first tabulate all $\mathrm{lca}_T(x, y)$ in quadratic time.

We have seen in Section 6.4 that at least some false positive edges are identified by good quartets. A convenient way of listing all good quartets Q in some BMG $(\vec{G}, \sigma)$ makes use of the *degree sequence* of $\vec{G}$, that is, the list $\alpha = ((\alpha_x^+, \alpha_x^-)|x \in V(\vec{G}))$ of pairs (α_x^+, α_x^-), where α_x^+ and α_x^- are the out- and the in-degree of the vertex $x \in V(\vec{G})$, and the list is ordered in positive lexicographical order. One easily checks that a good quartet contains neither a *2-switch* nor an *induced 3-cycle*, hence Q is uniquely defined by its degree sequence $((2,1),(2,3),(2,3),(2,1))$ as a consequence of [35, Thm. 1]. Regarding the coloring, it suffices to check that the two endpoints, that is, the vertices with indegree 1, have the same color $\sigma(u) = \sigma(x)$. This already implies $\sigma(v), \sigma(w) \neq \sigma(u) = \sigma(x)$. Since there is an edge between v and w, we also have $\sigma(v) \neq \sigma(w)$, i.e., the colors are determined up to a permutation of colors. The false positive edge is the one connecting the two vertices with outdegree 3.

For each reconciliation scenario (T, S, μ), all good quartets in the BMG $(\vec{G}, \sigma)$ are identified and the middle edge of the corresponding P_4 is then deleted from the RBMG (G, σ). The resulting graph will be referred to as (G_4, σ_4).

6.5.2 *Duplication/Loss Scenarios*

In order to assess the practical relevance of co-RBMGs, the abundance of non-cograph components in the simulated RBMGs has been measured. More precisely, for each simulated RBMG, the connected components of its restrictions to any three distinct colors have been determined and it has been tested whether these components are cographs, graphs of Type (B), or graphs of Type (C). In order to identify these graph types, algorithms of [108] have been applied to first identify an induced P_4 corresponding to a good quartet. If one exists, it has been checked for the existence of an induced P_5 and then tested whether its endpoints are connected, thus forming a hexagon characteristic for the a Type (C) graph. Otherwise, the presence of the P_4 implies Type (B), while the absence of induced P_4s guarantees that the component is a cograph.
As a direct implication of the following result, none of these connected components can be expected to be of Type (C).

Lemma 6.12. *Let (G, σ) be a connected 3-RBMG containing the induced C_6 $\langle x_1y_1z_1x_2y_2z_2\rangle$ of the form (r, s, t, r, s, t) for distinct colors r, s, and t, let (T, σ) be a tree explaining (G, σ), and set $v := \mathrm{lca}_T(x_1, x_2, y_1, y_2, z_1, z_2)$. Then there exist distinct $v_1, v_2, v_3 \in \mathsf{child}(v)$ such that either $x_1, y_1 \preceq_T v_1$, $x_2, z_1 \preceq_T v_2$, $y_2, z_2 \preceq_T v_3$ or $y_1, z_1 \preceq_T v_1$, $x_2, y_2 \preceq_T v_2$, $x_1, z_2 \preceq_T v_3$. In particular, T is not binary.*

Proof. Note that it suffices to show the first statement since this, in particular, implies that v has more than two children, thus T cannot be binary.
If $|V(G)| > 6$, then, due to the connectedness of G, at least one of the six vertices in the induced C_6 is adjacent to more than one vertex of one of the colors r, s, t, hence the first statement immediately follows from Lemma 5.38(iii).
Now consider the special case $|V(G)| = 6$. By Cor. 5.12, $\vec{G}(T, \sigma)$ contains a good quartet. W.l.o.g. let $\langle x_1y_1z_1x_2\rangle$ be a good quartet, thus $(x_1, z_1), (x_2, y_1) \in E(\vec{G})$ and $(z_1, x_1), (y_1, x_2) \notin E(\vec{G})$. This, in particular, implies $\mathrm{lca}_T(x_2, z_1) \prec_T$

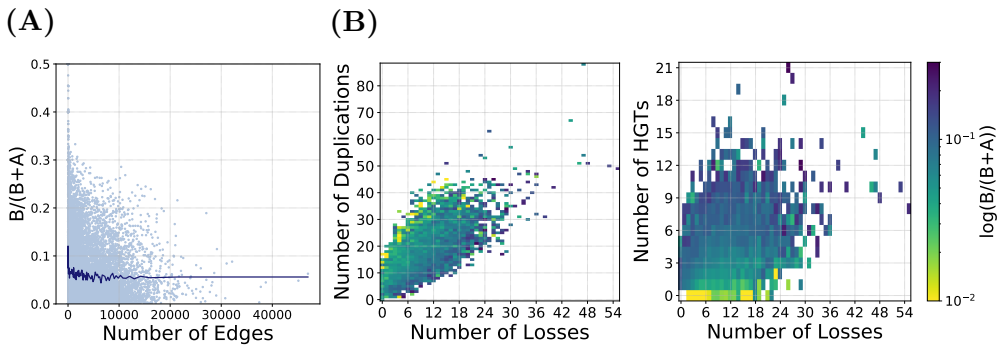


Fig. 34. Relative abundance $\eta = \frac{B}{B+A}$ of (B)-RBMGs in the simulation data. Panel (A) shows the dependence on the number of edges in the RBMG in every simulated scenario, and its average depicted by the line in darker blue. Scatter plots (B) show the dependence of η on the number of duplications and losses, and HGTs and losses, respectively.

$\mathrm{lca}_T(x_1, z_1)$, thus there are distinct children $v_1, v_2 \in \mathsf{child}(v)$ such that $x_1 \preceq_T v_1$ and $x_2, z_1 \preceq_T v_2$. Moreover, as $x_1y_1 \in E(G)$ and $(y_1, x_2) \notin E(\vec{G})$, we have $\mathrm{lca}_T(x_1, y_1) \prec_T \mathrm{lca}_T(x_2, y_1)$, hence $y_1 \preceq_T v_1$. Now consider y_2. Since $x_1y_2 \notin E(G)$ and $x_2y_2 \in E(G)$, it must hold $\mathrm{lca}_T(x_2, y_2) \preceq_T \mathrm{lca}_T(x_1, y_2)$, hence $y_2 \notin L(T(v_1))$. Assume, for contradiction, that $y_2 \preceq_T v_2$. Then, as $y_2z_2 \in E(G)$ and $\mathrm{lca}_T(y_2, z_1) \preceq_T v_2$, we clearly have $z_2 \preceq_T v_2$. However, this implies $\mathrm{lca}_T(x_2, z_2) \prec_T \mathrm{lca}_T(x_1, z_2)$, contradicting $x_1z_2 \in E(G)$. We therefore conclude that there must exist a vertex $v_3 \in \mathsf{child}(v) \setminus \{v_1, v_2\}$ such that $y_2 \preceq_T v_3$. One easily checks that this implies $z_2 \preceq_T v_3$, which completes the proof. □

In particular, therefore, we have

Corollary 6.4. *If (T, σ) is a binary leaf-labeled tree, then $G(T, \sigma)$ does not contain a connected component of Type* (C).

Following our expectations, not a single Type (C) component has been encountered in 14,000 simulated scenarios. Although events that generate more than two offspring lineages are logically possible in real data, most multifurcation in phylogenetic trees are considered to be "soft polytomies", arising from insufficient data. Various biologically appropriate approaches have been proposed to obtain fully resolved, binary trees from trees containing soft polytomies [187, 136, 197]. Type (C) 3-RBMGs thus should be very unlikely under biologically plausible assumptions on the model of evolution. Here, only the abundance of Type (B) components relative to all Type (A) and Type (B) components is considered. Their ratio is denoted by η. The results are summarized in Fig. 34. It turns out that η is usually below 20% and increases with the number of loss and HGT events. More precisely, 83.47% of the 14,000 scenarios have at least one Type (B) component and 16.53% do not have Type (B) components at all. Among all 3-colored connected components taken from the restrictions to any three colors, 94.41% are of Type (A) and 5.59% are of Type (B).

A graph G is called P_4-sparse if every induced subgraph on five vertices contains at most one induced P_4 [117]. The interest in P_4-sparse graphs derives

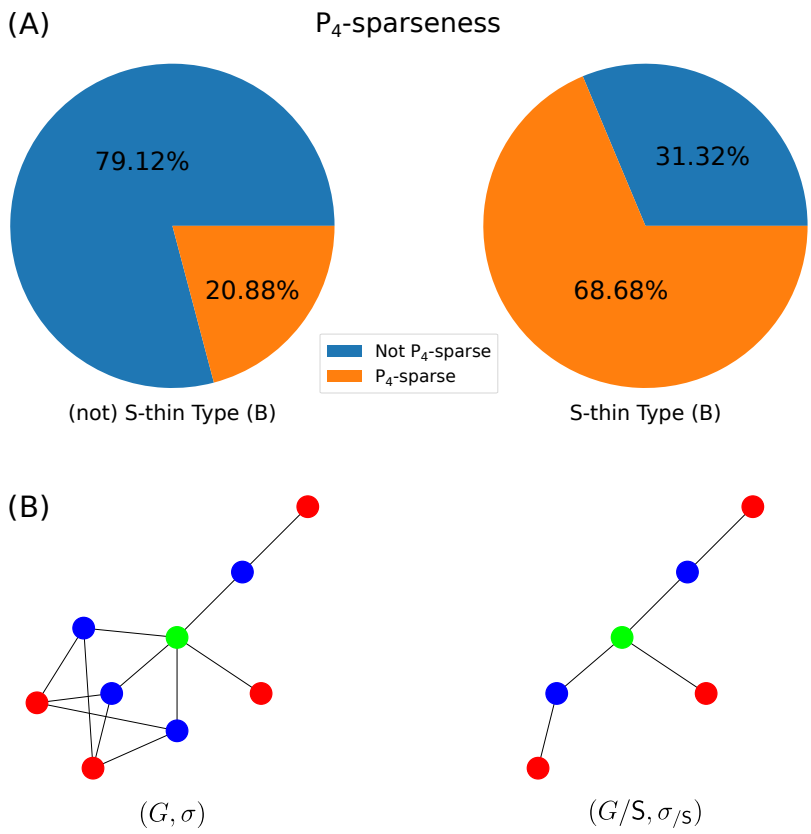


Fig. 35. *Top:* Among the 14,000 simulated scenarios a majority of 79.12% of the (not necessarily S-thin) 3-colored Type (B) components are not P_4-sparse. For the corresponding S-version of those 3-colored components only 31.32% are not P_4-sparse while 68.68% are P_4-sparse. *Below:* One of the simulated 3-colored Type (B) components (G, σ), which is not S-thin, and its corresponding S-thin version $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$.

from the fact that the cograph editing problem is solvable in linear time from P_4-sparse graphs [150]. It is of immediate practical interest, therefore, to determine the abundance of P_4-sparse RBMGs that are not cographs. Among the 14,000 simulated scenarios, it has been found that about 20.9% of the 3-colored Type (B) components are P_4-sparse, while the majority contains "overlapping" P_4s. Next, the corresponding S-thin graphs have been considered. Somewhat surprisingly, this yields a reversed situation, where more than two thirds of the S-thin 3-colored Type (B) components are now P_4-sparse, while only a minority of 31.32% is not P_4-sparse. An example of an undirected colored graph (G, σ) and its corresponding S-thin version $(G/\mathsf{S}, \sigma_{/\mathsf{S}})$, which has been found during these simluations, is shown in Panel (B) of Fig. 35.

The next step was an investigation of the relationship of the RBMG $G(T, \sigma)$ and the orthology graph Θ (see Fig. 36). It has been empirically confirmed that $E(\Theta) \subseteq E(G(T, \sigma))$ in the absence of HGT (not shown). Also following our expectations, the fraction $|E(G(T, \sigma)) \setminus E(\Theta)|/|E(G(T, \sigma))|$ of false positive orthology predictions in an RBMG is small as long as duplications and losses remain moderate (l.h.s. panel in Fig. 36). Most of the false positive or-

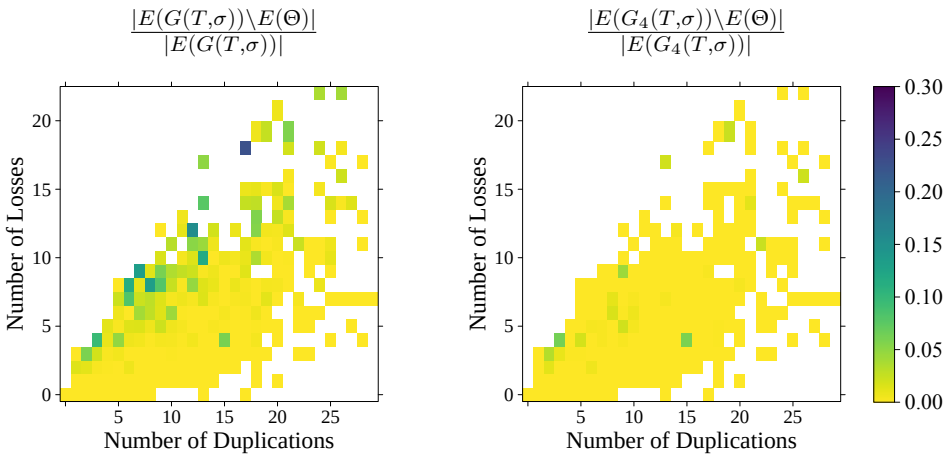


Fig. 36. Fraction of non-orthology edges in the reciprocal best match graph (l.h.s.). The x-axis, resp., y-axis indicate the total number of duplications, resp., losses in the simulated scenarios. Most of the false positive orthology assignments in the l.h.s. panel are removed by deleting the middle edge of good quartets (r.h.s. panel). White background indicates *no data*.

thology calls are associated with large numbers of losses for a given number of duplication.

It finally turns out that good quartets eliminate nearly all false positive edges from the RBMG and leave a nearly perfect orthology graph (r.h.s. panel in Fig. 36). As we have seen so far, reciprocal best matches indeed form an excellent approximation of orthology in Duplication/Loss (DL) scenarios. In particular, the good quartets identify nearly all false positive edges, making it easy to remove the few remaining P_4s using a generic cograph editing algorithm [150].

6.5.3 *Evolutionary Scenarios with Horizontal Gene Transfer*

The benign results above beg the question how robust they are under HGT. Gene family histories with HGT have been a topic of intense study in recent years [56, 224, 18, 173]. Following the so-called DTL-scenarios as proposed e.g. by Tofigh et al. [224], Bansal et al. [18], we relax the notion of reconciliation maps since ancestry is no longer preserved. More precisely, we replace Axiom (R2) by

(R2w) *Weak Ancestor Preservation.*
If $x \prec_T y$, then either $\mu(x) \preceq_S \mu(y)$ or $\mu(x)$ and $\mu(y)$ are incomparable w.r.t. $\prec_S$.

and add the following constraints

(R3.iii) *Addition to the Speciation Constraint.*
If $\mu(x) \in W^0$, then $\mu(v) \preceq_T \mu(x)$ for all $v \in \mathsf{child}(x)$.

(R4) *HGT Constraint.*
If x has a child y such that $\mu(x)$ and $\mu(y)$ are incomparable, then x also has a child y' with $\mu(y') \preceq_S \mu(x)$.

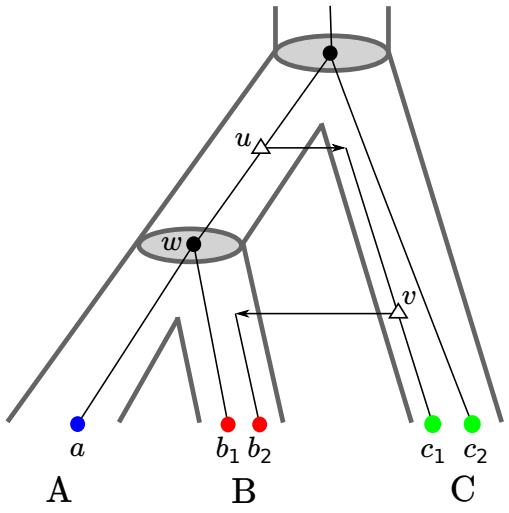


Fig. 37. A gene tree $(T, t_T, \lambda, \sigma)$ reconciled with a species tree S. Here, we have two transfer edges uv and vb_2 with $t_T(u) = t_T(v) = \triangle$. For the two children w and v of u it holds $\sigma(L(T(w))) \cap \sigma(L(T(v))) \neq \emptyset$, a property that is shared with duplication vertices. For the two children b_2 and c_1 of v it holds $\sigma(L(T(b_2))) \cap \sigma(L(T(c_1))) = \emptyset$, a property that is shared with speciation vertices. In this example, c_1 and c_2 are xeno-orthologs and the pairs $(c_1, c_2), (c_2, c_1)$ will be excluded from the resulting orthology relation.

Property (R2w) equivalently states that, if $x \prec_T y$, then we must not have $\mu(y) \prec_S \mu(x)$, which would invert the temporal order. Property (R3.iii) (which follows from (R2) but not from (R2w)) ensures that the children of speciation events are still mapped to positions that are comparable to the image of the speciation node. Condition (R4), finally, requires that every horizontal transfer event also has a vertically inherited offspring. Note that Condition (R4) is void if (R2) holds. In summary the Axioms (R0), (R1), (R2w), (R3.i), (R3.ii), (R3.iii), and (R4) are a proper generalization of Def. 6.1. However, these axioms are not sufficient to ensure time consistency (see [173] for details). This choice of axioms also rules out some scenarios that may appear in reality (or simulations) but which are not observable when only evolutionary divergence is available as measurement. For example, Condition (R3.ii) excludes scenarios in which HGT events have no surviving vertically inherited offspring.

Furthermore, the event labeling map t_T can be extended to include HGT as an additional event type denoted by the symbol $\triangle$. We define $t_T : V(T) \to \{\circledcirc, \odot, \bullet, \square, \triangle\}$ such that $t_T(u) = \triangle$ if and only if u has a child v such that $\mu(u)$ and $\mu(v)$ are incomparable. Since the offspring lineages of an HGT event are not equivalent, it is useful to introduce an edge labeling $\lambda : E(T) \to \{0, 1\}$ such that $\lambda(uv) = 1$ if $\mu(u)$ and $\mu(v)$ are incomparable w.r.t. $\prec_S$, i.e., uv corresponds to a transfer edge. This edge labeling will be investigated in detail in Chapter 8 as the basis of Fitch's xenology relation. Alternatively, the asymmetry can be handled by enforcing an ordering of the vertices, see [100].

Evolutionary scenarios with horizontal transfer may lead to a situation where two genes x, y in the same species, i.e., with $\sigma(x) = \sigma(y)$, derive from a speciation, i.e., $\mathrm{lca}_T(x, y) = \bullet$. This is the case when the two lineages underwent an HGT event that transferred a copy back into the lineage in which the other gene has been vertically transmitted. We call such genes *xeno-orthologs* and exclude them from the orthology relation, see Fig. 37. This choice is motivated (1) by the fact that, by definition, genes of the same species cannot be recognized as reciprocal best matches, and (2) from a biological perspective they behave

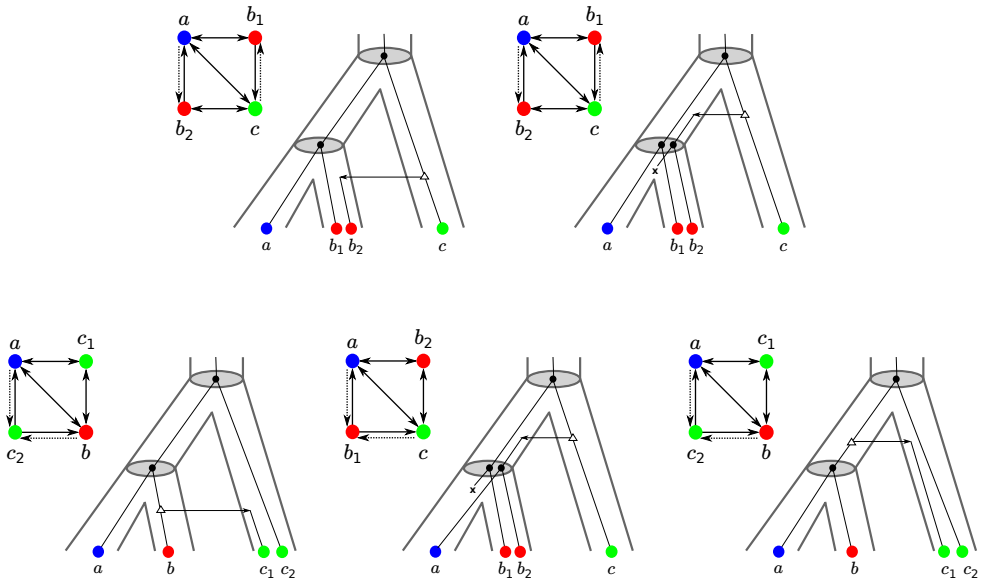


Fig. 38. Scenarios with four genes, three species, and a single HGT event. The BMG is shown for each scenario. The extra dashed arrows in the BMG represent true orthology relations that are missed because of the HGT event. The first two cases (on the top) introduce an induced P_4 into the RBMG, which may potentially serve as indication for HGT events. The remaining three cases (bottom) yield $K_3 \cup K_1$ instead of K_4. This situation would be missed by methods based on cograph editing.

rather like paralogs. In scenarios with HGT we therefore modify the definition of the orthology graph such that $E(G_1 \triangledown G_2)$ is replaced by

$$E(G_1 \tilde{\triangledown} G_2) := E(G_1) \cup E(G_2) \cup \{uv \mid u \in V(G_1), v \in V(G_2), \sigma(u) \neq \sigma(v)\}. \tag{21}$$

The extremal map $\hat{t}_T$ as in Def. 6.4 cannot easily be extended to include HGT as the events ● and □ on some vertex u are solely defined on two exclusive cases: either $\sigma(L(T(u_1)))$ and $\sigma(L(T(u_2)))$ are disjoint or not for $u_1, u_2 \in \mathsf{child}(u)$. Both cases, however, can also appear when HGT events are involved (see Fig. 37 for an example). That is, the fact that $\sigma(L(T(u_1)))$ and $\sigma(L(T(u_2)))$ are disjoint or not, does not help to unambiguously identify the event types in the presence of HGT.

Prop. 6.1 can be generalized to the case that $(T, t_T, \lambda, \sigma)$ contains HGT events. The existence of reconciliation maps from an event-labeled tree $(T, t_T, \lambda, \sigma)$ to an *unknown* species tree can be characterized in terms of species triples $\sigma(a)\sigma(b)|\sigma(c)$ that can be derived from $(T, t_T, \lambda, \sigma)$ as follows: Denote by $\mathcal{E} := \{e \in E(T, t_T, \lambda, \sigma) \mid \lambda(e) = 1\}$ the set of all transfer edges in the labeled gene tree and let $(T_{\overline{\mathcal{E}}}, t_T, \sigma)$ be the forest obtained from $(T, t_T, \lambda, \sigma)$ by removing all transfer edges. By definition, $\mu(x)$ and $\mu(y)$ are incomparable for every transfer edge xy in T. The set $\mathcal{S}(T, t_T, \lambda, \sigma)$ is the set of triples $\sigma(a)\sigma(b)|\sigma(c)$ where $\sigma(a)$, $\sigma(b)$, $\sigma(c)$ are pairwise distinct and either

1. $ab|c$ is a triple displayed by a connected component T' of $T_{\overline{\mathcal{E}}}$ such that the root of the triple is a speciation event, i.e., $t_T(\mathrm{lca}_{T'}(a, b, c)) = ●$, or

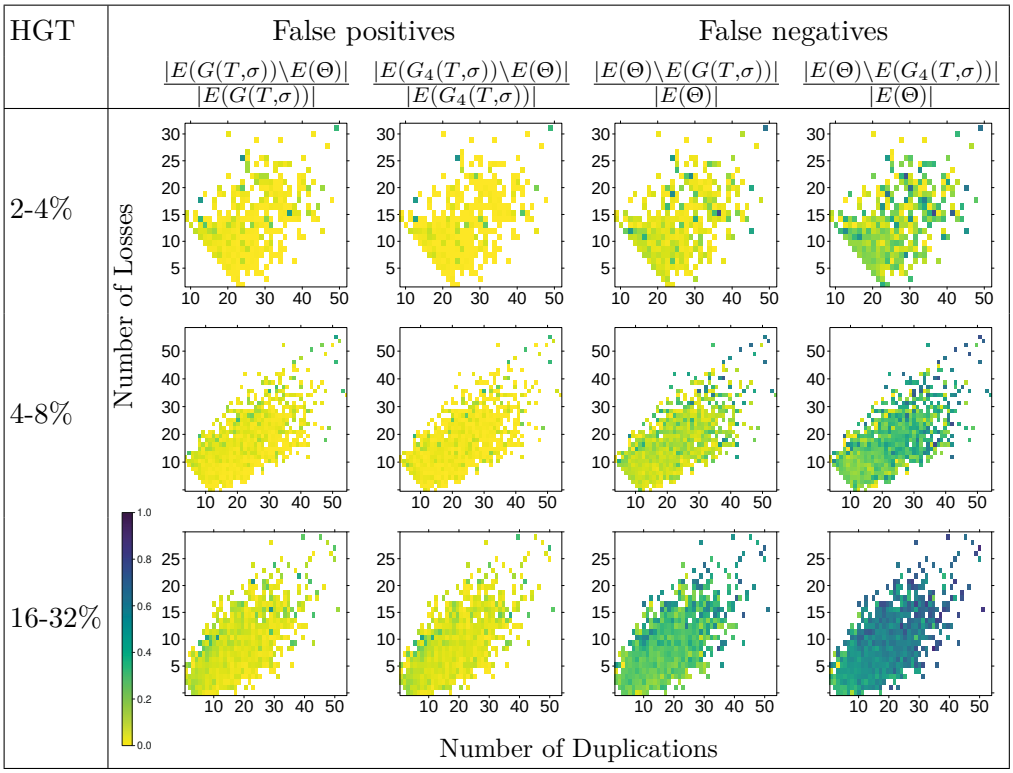


Fig. 39. Dependence of the fraction of false positive and false negative orthology assignments in RBMGs in the presence of different levels of HGT, measured as percentage of HGT events among all events in the simulated true gene trees $\tilde{T}$. As in Fig. 36, data are shown as functions of the number of duplication and loss events in the scenario. While the number of false positives seems to depend very little on even high levels of HGT, the fraction of false negatives is rapidly increasing. Since HGT introduces good quartets that comprise only true orthology edges, their removal further increases the false positive rate (last column).

2. $a, b \in L(T_{\overline{\mathcal{E}}}(x))$ and $c \in L(T_{\overline{\mathcal{E}}}(y))$ for some transfer edge xy or yx of T.

Proposition 6.2. *[99] Given an event-labeled, leaf-labeled tree (T, t_T, σ), there is a reconciliation map $\mu : V(T) \to V(S) \cup E(S)$ to some species tree if and only if $\mathcal{S}(T, t_T, \sigma)$ is consistent. In this case, (T, t_T, σ) can be reconciled with every species tree S that displays the triples in $\mathcal{S}(T, t_T, \sigma)$.*

Here, we have not added additional constraints on reconciliation maps that ensure that the map is also "time-consistent", that is, genes do not travel "back" in the species tree, see [173] for further discussion on this. However, Prop. 6.2 gives at least a necessary condition for the existence of time-consistent reconciliation maps. A simple proof of Prop. 6.2 for the case that T is binary and does not contain HGT events can be found in [105]. Moreover, generalizations of reconciling event-labeled gene trees with species networks have been established by [103].

In contrast to pure DL scenarios, it is no longer guaranteed that all true orthology relationships are also reciprocal best matches. Fig. 38 gives counterexamples. In three of these scenarios the RBMG contains an induced P_4

that mimics a good quartet. Removal of the middle edge of good quartets therefore not only reduces false positives in DL scenarios but also introduces additional false negatives in the presence of HGT. This is also reflected by the simulation scenarios with HGT (see Fig. 39).

6.6 SUMMARY

The theoretical part of this chapter clarifies the relationships between (reciprocal) best match graphs, orthology, reconciliation map, gene tree, species tree, and event map for the case of DL scenarios. The orthology graph Θ is necessarily a subgraph of the RBMG. In the absence of HGT, RBMGs therefore produce only false positive but no false negative orthology assignments. Using not only reciprocal best matches but all best matches, furthermore, shows that good quartets identify almost all false positive edges. Simulations confirm that removing the central edge of all good quartets in the reciprocal best match graph yields nearly perfect orthology estimates. This, however, implies that orthology inference is not solely based on reciprocal best matches. Instead, it is necessary to also include certain directional best matches, namely those that identify good quartets.

7

FROM BEST HITS TO BEST MATCHES

The previous chapters were concerned with the relationship of (reciprocal) best match graphs and gene trees, in particular they provided characterizations for BMGs and RBMGs as well as algorithms for the reconstruction of least resolved trees from the underlying BMG/RBMG. Moreover, it has been shown that, in the absence of HGT events, these insights can be successfully used to identify almost all false positive orthology assignments in the RBMG. Up to this point, however, we did not treat the question how to initially recover the best match relation from data. Many of the commonly used methods for orthology detection start from pairwise best (`blast`) hits as an approximation for evolutionary most closely related pairs of genes. This approximation becomes exact for ultrametric dissimilarities, i.e., under the Molecular Clock Hypothesis but it fails in general whenever there are large lineage specific variations of the evolutionary rate among paralogous genes. In this chapter we ask to what extent the knowledge of an additive evolutionary distance can be leveraged to determine the best match relation. To this end, we investigate the theoretical connection and the impact of the missing piece of information, i.e., the exact position of the root.

We start in Section 7.1 with a discussion about the relationship of additive metrics and dissimilarity measures. Section 7.2 shows how additive metrics can be transformed into quartets in general and, in particular, how quartets can be estimated from sequence data. Quartets with known outgroups are then used in Section 7.3 to identify best matches. The question to what extent outgroups can be reliably identified is treated in Section 7.4.

The results of this chapter have only recently been submitted to *23th Conference on Algorithmic Computational Biology (RECOMB 2019)* [211].

7.1 ADDITIVE METRICS AND DISSIMILARITY MEASURES

The lca function of a phylogenetic tree cannot be measured directly from data but has to be inferred through the comparative analysis of the data representing the leaf set. Likewise, the best match relation has to be inferred indirectly from measurable quantities. Conceptually, best matches are closely related to *best hits* (e.g. in `blast` searches) and ways of estimating *most similar* or *least dissimilar* sequences. Denote by $\ell : E(T) \to \mathbb{R}^+$ an assignment of positive lengths to the edges of a planted phylogenetic tree T with planted root 0_T and leaf set L, which we interpret as a measure proportional to the number of evolutionary events. It gives rise to a metric distance function $d_{T,\ell}(x, y)$ on L defined as the sum of the lengths $\ell(e)$ of the edges e along the unique path connecting the leaves x and y in T. From T we obtain an associated *unrooted* tree $\overline{T}$ by (i) omitting the planted root 0_T and its incident edge, and (ii), in case the root ρ in T has exactly two children u_1 and u_2, by replacing the path

$u_1\rho u_2$ by a single edge u_1u_2 with length $\ell(u_1u_2) := \ell(u_1\rho) + \ell(\rho u_2)$. Note that the dissimilarity function ℓ is by construction the same on T and $\overline{T}$. Thus $\overline{T}$ determines T up to the position of the root, i.e., T is obtained from $\overline{T}$ by inserting the root into an edge of $\overline{T}$ or declaring an inner vertex of $\overline{T}$ as the root.

The problem of determining the position of the root in an unrooted tree $\overline{T}$ has been well studied in the phylogenetic literature [129]. The most common approach is the inclusion of an outgroup, i.e., a taxon z known to branch earlier than the taxa of interest. The root is then located in the branch leading to z. Outgroup rooting can be unreliable in the presence of rapid radiations or when only very distant outgroups are available [109, 205]. The simplest method is midpoint rooting [216], which places the root at the midpoint on the longest path in the tree. Despite its simplicity it often works remarkably well [106]. An interesting variation on this theme is minimum variance rooting [155]. The estimation of dated phylogenies using a relaxed clock assumption yields an estimate for the position of the root as a by-product [58]. A related Bayesian method was introduced in [111]. In a phylogenomics setting, the root of the species tree can also be obtained by minimizing the number of inferred gene duplications [124]. Most recently, non-reversible substitution models have been employed for estimating rooted phylogenic trees [232, 34].

A dissimilarity d on L is called *additive* if there is an unrooted tree $\overline{T}$ with edge lengths ℓ such that $d = d_{\ell,\overline{T}}$. A key result in mathematical phylogenetics [206, 28] characterizes additive (pseudo)metrics as those that satisfy the *four point condition*. It states that d is additive if and only if the restriction of d to each subset L' of L with $|L'| = 4$, usually called a *quartet*, is additive and thus, determines a tree on four leaves. Furthermore, the unrooted tree $\overline{T}$ is uniquely defined by d. In principle, therefore, distance data completely determines a phylogenetic tree up to the position of the root.

In Chapter 4 we already discussed that the evolutionary relatedness of two extant genes x and y can be expressed by their divergence time $\tau(x, y) = 2\hat{\tau}(\mathrm{lca}(x, y))$, where $\hat{\tau}$ is the age of $\mathrm{lca}(x, y)$ in the corresponding gene tree. In particular, based on the fact that divergence times are by definition ultrametric, the best match relation can be defined in terms of the divergence time, i.e., $x \to y$ if and only if

$$y \in \arg\min_{y' \in L[t]} \tau(x, y') \tag{22}$$

This equation is restated at that point from Chapter 4 for latter reference in the current chapter.

Divergence times cannot be measured in most cases, however, since this would require the knowledge of dated last common ancestors. The next best choice is the evolutionary distance, which measures the number of evolutionary events that have taken place. For each edge $e = uv$ in T it is given by $\ell(e) = \int_{\hat{\tau}(u)}^{\hat{\tau}(v)} \mu_e(t)dt$, where $\mu_e(t)$ is the rate of evolution. In general μ_e depends both on the lineage, and thus the individual edges in T, as well as on the exact point in time along e. It associates with each edge e a measure $\ell(e)$ of changes incurred, and thus an additive distance. If $\mu_e(t) = \mu_0$ is constant, we simply have

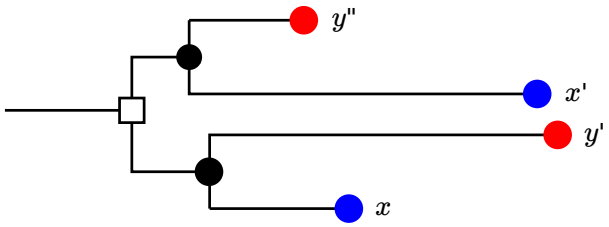


Fig. 40. Lineage-specific rate variation between paralogs. The gene tree, with branch length indicating an additive evolutionary distances, comprises two paralogs arising from a duplication (□) that predates the speciation (●) of the red and blue species. We have $\mathrm{lca}(x, y') \prec \mathrm{lca}(x, y'')$ but $d(x, y'') < d(x, y')$.

$d_{\ell,T}(x, y) = \mu_0 \tau(x, y)$. This corresponds to the Molecular Clock Hypothesis. As a consequence, genes with minimal genetic distance are not necessarily best matches in the sense of evolutionary relatedness; Fig. 40 shows a paradigmatic example arising from lineage-specific rate differences between paralogs.

It is worth noting that additive evolutionary distances are not directly accessible from data. While it is easy to obtain dissimilarities $d'(x, y)$ on L from pairwise alignments, d' is usually a systematic under-estimate of the number of events d due to back-mutations, and thus not additive. In practice, the conversion of measurements of d' into an additive distance d that quantifies the number of evolutionary events is based on a Markov model. For sequence data, this may be the Jukes-Cantor model [122] or one of its more elaborate variants [128, 90, 218]. In the most benign setting, d and d' are related by a monotone transformation. If, in addition, μ is constant, then both d and d' could be substituted for τ in Equ. (22) to determine best matches. However, violations of the clock hypothesis usually make this a poor approximation. In general it is not possible to estimate the correct tree topology from a non-additive metric [192]. Therefore it is not possible to avoid the transformation from measured dissimilarities d' to the estimated evolutionary distance d.

7.2 Additive Metrics and Quartets

This section is concerned with the relationship between additive metrics and quartets. In particular, it is shown how quartets can be estimated from sequence alignments using concepts from statistical geometry.

Consider an unrooted tree $\overline{T}$ with leaf set L. For any four distinct leaves $p, q, r, s \in L$ denote by $\overline{T}[p, q, r, s]$ the unrooted tree obtained by suppressing all vertices of degree 2 in the union of the paths in $\overline{T}$ that connect p, q, r, s. We write $(pq|rs)$ if there is an edge e in $\overline{T}$ so that $\{p, q\}$ and $\{r, s\}$ are in different connected components of the forest obtained by removing e from $\overline{T}$. This quartet relation [196, 65] can then be expressed equivalently as $(pq|rs)$ if and only if

$$d(p, q) + d(r, s) < d(p, r) + d(q, s), d(p, s) + d(q, r)\,. \tag{23}$$

In fact, for additive metrics, the two distance sums on the r.h.s. are equal [206, 28]. All three terms are equal if and only if the four points form a star,

whence the existence of a separating edge requires the strict inequality. By a slight abuse of notation, we write $\overline{T}[p,q,r,s] = (pq|rs)$ if Equ. (23) holds, and $\overline{T}[p,q,r,s] = \times$ if no quartet exists on these four leaves, i.e., if $\overline{T}[p,q,r,s]$ is the star tree.

7.2.1 *Estimation of quartets from sequence data*

Several different approaches to estimate quartets from aligned sequence data have been discussed in the literature. Most directly, Equ. (23) can be used to determine the dominating triple on $\{p,q,r,s\}$ directly from the additive distance d. A weight can be assigned to the triple by setting $w(pq|rs) = (1 - d_0/d_2)\exp(d_1 - d_2)$ where d_0, d_1, and d_2 are the distance sums in Equ. (23) ordered by increasing value, i.e., such that $d_0 \leq d_1 \leq d_2$ [15]. This type of approach, however, requires the prior transformation of sequence differences to the additive distance d, or the estimate of d directly from the sequence using a suitable model of sequence evolution. The latter is explored with the so-called Likelihood Mapping method [213].

A more elegant approach assumes a multiple alignment of some sequences x, y', y'', and z, which we assume to appear in this order. Following the idea of statistical geometry [61, 171], each alignment column belongs to one of the 15 categories determined by which of the four sequences x, y', y'', and z feature the same character:

	$\mathfrak{C}1$	$\mathfrak{C}2$	$\mathfrak{C}3$	$\mathfrak{C}4$	$\mathfrak{C}5$	$\mathfrak{C}6$	$\mathfrak{C}7$	$\mathfrak{C}8$	$\mathfrak{C}9$	$\mathfrak{C}10$	$\mathfrak{C}11$	$\mathfrak{C}12$	$\mathfrak{C}13$	$\mathfrak{C}14$	$\mathfrak{C}15$
x	a	a	a	a	b	a	a	a	a	a	a	b	b	b	a
y'	a	a	a	b	a	a	b	b	a	b	b	a	a	c	b
y''	a	a	b	a	a	b	a	b	b	a	c	a	c	a	c
z	a	b	a	a	a	b	b	a	c	c	a	c	a	a	d

The categories $\mathfrak{C}1$ through $\mathfrak{C}5$ and $\mathfrak{C}15$ do not convey phylogenetic information. Of the remaining ones, $\mathfrak{C}6$, $\mathfrak{C}9$, and $\mathfrak{C}14$ support $(xy'|y''z)$, $\mathfrak{C}7$, $\mathfrak{C}10$, and $\mathfrak{C}13$ support $(xy''|y'z)$, and $\mathfrak{C}8$, $\mathfrak{C}11$, and $\mathfrak{C}12$ support $(xz|y'y'')$ [172]. Denoting by d_{aaaa}, etc., the number of alignment columns belonging to a given category, the support scores for *geometry mapping* [172] are

$$\begin{aligned} S(xy'|y''z) &= d_{aabb} + \frac{1}{2}(d_{aabc} + d_{bcaa}) \\ S(xy''|y'z) &= d_{abab} + \frac{1}{2}(d_{abac} + d_{baca}) \\ S(xz|y'y'') &= d_{abba} + \frac{1}{2}(d_{abca} + d_{baac}) \end{aligned} \tag{24}$$

Using $S := S(xy'|y''z) + S(xy''|y'z) + S(xz|y'y'')$, normalized scores are defined as $s(xy'|y''z) := S(xy'|y''z)/S$. This unweighted version can be extended to a weighted version when a non-trivial distance measure D on the underlying alphabet is given. As derived in [172], a support value for the three possi-

ble quartets can be computed separately for each alignment column i as the isolation index for the distances on the four characters:

$$\begin{aligned} 2\beta_i(xy'|y''z) &= D_i^* - (D(x_i, y_i') + D(y_i'', z_i)) \\ 2\beta_i(xy''|y'z) &= D_i^* - (D(x_i, y_i'') + D(y_i', z_i)) \\ 2\beta_i(xz|y'y'') &= D_i^* - (D(x_i, z_i) + D(y_i', y_i'')) \end{aligned} \tag{25}$$

Here, D_i^* is the largest of the three distance sums appearing in Equ. (23). Summing up the $\beta_i(\,.\,)$ values over all alignment columns i yields aggregated support scores $\beta(\,.\,)$. These are conveniently normalized to relative values as in the unweighted case. If no quartet can be inferred unambiguously, then we revert to the assumption $\mathrm{lca}(x, y') = \mathrm{lca}(x, y'')$. The quartet mapping approach is particularly appealing because the computation of the quartet support values is simple, can be performed efficiently, and does not require a particular model of sequence evolution.

7.3 FROM QUARTETS TO ROOTED TRIPLES

The idea of this section is to use quartets with a known outgroup in order to infer rooted triples, which are then used to retrieve the best matches. We present a workflow (Algorithm 7) and discuss under which conditions the set of best matches can be correctly identified.

The most common method to specify the root of a phylogenetic tree is the use of so-called outgroups, that is, additional taxa that are known *a priori* to be outside a monophyletic group of interest. Given a planted (or rooted) phylogenetic tree, on the other hand, monophyletic groups are the leaf sets of a subtree, i.e., L' is a monophyletic group if and only if there is a vertex $u \in V(T)$ such that $L' = L(T(u))$. Every leaf $x \in L \setminus L'$ is an outgroup for L'.

Every edge in an unrooted tree $\overline{T}$ defines a split $L'|L''$ of L, where L' and L'' are the leaves in the connected components of $\overline{T} \setminus e = \overline{T'} \cup \overline{T''}$. At most one of the two subtrees $\overline{T'}$ and $\overline{T''}$ contains the root of the underlying phylogenetic tree T. If the root is not contained in $\overline{T'}$, then the tree $T' = \overline{T'} \cup \{e\}$ that is planted at the endpoint of e, describes a monophyletic group. In this case all $x \in L''$ are outgroups for T'. Which subtrees of $\overline{T}$ correspond to monophyletic groups is determined by the position of the root and therefore, requires external information.

It will be convenient in the following to define outgroups not only for monophyletic groups.

Definition 7.1. *For a phylogenetic tree T with leaf set L consider a subset $L' \subseteq L$ and a leaf $z \in L \setminus L'$. We say that z is an* outgroup *for L' if* $\mathrm{lca}(L') \prec \mathrm{lca}(L', z)$.

Let us now return to the quartets of $\overline{T}$. The following simple result, illustrated in Fig. 41, shows that quartets can be used to infer inequalities between lca vertices in T, provided one of the four leafs is known to be an outgroup for the other three:

Lemma 7.1. *Suppose z is an outgroup for $\{x, y', y''\}$ in T. If $\overline{T}[x, y', y'', z]$ is fully resolved, then*

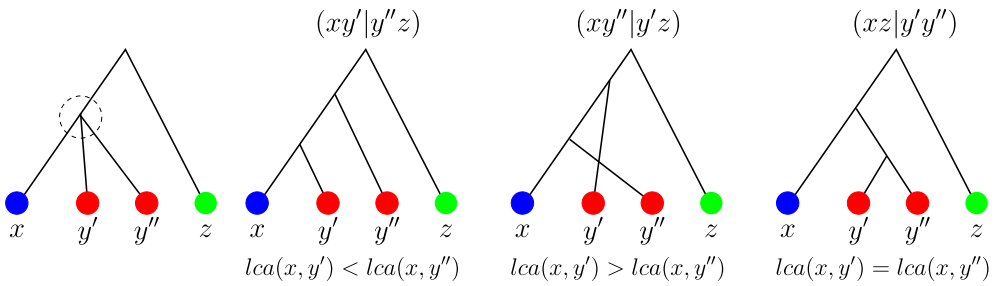


Fig. 41. Relation of the last common ancestors $\mathrm{lca}(x, y')$ and $\mathrm{lca}(x, y'')$, resp., with quartets on $\{x, y', y'', z\}$ with a trusted outgroup z.

(i) $\mathrm{lca}(x, y') = \mathrm{lca}(x, y'')$ *if and only if* $\overline{T}[x, y', y'', z] = (xz|y'y'')$,

(ii) $\mathrm{lca}(x, y') \prec \mathrm{lca}(x, y'')$ *if and only if* $\overline{T}[x, y', y'', z] = (xy'|y''z)$, *and*

(iii) $\mathrm{lca}(x, y') \succ \mathrm{lca}(x, y'')$ *if and only if* $\overline{T}[x, y', y'', z] = (xy''|y'z)$.

Otherwise, $\overline{T}[x, y', y'', z] = \times$ *and* $\mathrm{lca}(x, y') = \mathrm{lca}(x, y'')$.

Proof. Since z is an outgroup by assumption, there are only three possible fully resolved rooted trees with $L = \{x, y', y'', z\}$, see Fig. 41. Each of these trees corresponds to a unique quartet (annotated at the top). The relationship between $\mathrm{lca}(x, y')$ and $\mathrm{lca}(x, y'')$ is determined by the tree topology. The statement follows by inspecting the three cases. If $\overline{T}[x, y', y'', z]$ is not fully resolved, no quartet is defined on $\{x, y', y'', z\}$, i.e., $\overline{T}$ is the star tree and thus $\mathrm{lca}(x, y') = \mathrm{lca}(x, y'') = \mathrm{lca}(y', y'')$. □

Observation 7.1. *If* $u' = \mathrm{lca}(x, y')$ *and* $v' = \mathrm{lca}(x, y'')$ *for* $x, y', y'' \in L$, *then* u' *and* v' *are comparable w.r.t.* $\preceq$ *in* T.

Lemma 7.1 together with Obs. 7.1 implies that quartets with known outgroups can be used to identify best matches. More precisely, in order to determine the set $\{y \in L[s] \mid x \to y\}$, it suffices to consider leaf sets $\{x, y', y'', z\}$ with $y', y'' \in L[s]$ such that z is an outgroup for $\{x, y', y''\}$. By Lemma 7.1, any set of this type implies an (in)equality between $\mathrm{lca}(x, y')$ and $\mathrm{lca}(x, y'')$. It may not be necessary to consider all quartets. To explore ways to reduce the computational effort, let us assume that for given $x \in L$ and $s \in S$, $s \neq \sigma(x)$, we can identify sets $\mathcal{Y} \subseteq L[s]$ and $\mathcal{Z} \subseteq L$ such that the following three assumptions are satisfied:

(A0) The noise in the data is small enough so that for any four taxa $\{x, y', y'', z\}$ with $y', y'' \in \mathcal{Y}$ and $z \in \mathcal{Z}$ one of the three possible quartets or the star topology is inferred correctly.

(A1) The candidate set $\mathcal{Y} \subseteq L[s]$ contains all best matches of x in species s (but usually also additional leaves).

(A2) $\mathcal{Z}$ is a non-empty set of outgroups for $\mathcal{Y} \cup \{x\}$.

Algorithm 7 Overall Workflow

Require: Reference vertex x

1: retrieve a sufficient set $\mathcal{Y} \subseteq L[s]$ of candidate best matches for x with color s
2: determine a set $\mathcal{Z}$ of outgroup vertices for $\mathcal{Y} \cup \{x\}$
3: initialize an edgeless digraph Γ with vertex set $\mathcal{Y}$
4: **for all** pairs $y', y'' \in \mathcal{Y}$ **do**
5: **for all** $z \in \mathcal{Z}$ **do**
6: determine significantly supported quartet on $\{x, y', y'', z\}$
7: determine consensus quartet over all choices of $z \in \mathcal{Z}$
8: **if** consensus quartet implies $\mathrm{lca}(x, y_1) \preceq \mathrm{lca}(x, y_2)$ **then**
9: insert the directed edge (y_2, y_1) into Γ
10: compute the strongly connected components of Γ
11: report strongly connected components without out-edges as the set of best matches $\{y \in \mathcal{Y} | x \to y\}$

The discussion so far suggests to use the workflow defined in Algorithm 7 to identify the best matches of x.

Lemma 7.2. *Algorithm 7 correctly identifies the set of best matches of x with color s as the unique strongly connected component of Γ without out-edges, provided assumptions* (A0)*,* (A1)*, and* (A2) *are satisfied.*

Proof. Assumptions (A1) and (A2) imply that comparison of the last common ancestors can be performed in terms of the quartets according to Lemma 7.1, which by assumption (A0) are all inferred correctly. Therefore Lines 4-7 compute all quartets correctly and thus, the inequality between $\mathrm{lca}(x, y_1)$ and $\mathrm{lca}(x, y_2)$ is inferred correctly. The auxiliary graphs Γ therefore contains at least one arc between any two vertices $y', y'' \in \mathcal{Y}$ and both the arc (y', y'') and (y'', y') if and only if $\mathrm{lca}(x, y') = \mathrm{lca}(x, y'')$, i.e., the strongly connected components are cliques. Since the $\mathrm{lca}(x, y)$, $y \in \mathcal{Y}$, are inner vertices of T that are totally ordered along the path from x to the root of T (Obs. 7.1), there is a unique strongly connected component B in Γ that has no out-edges and whose vertices are those $y \in B$ for which $\mathrm{lca}(x, y)$ is minimal. Thus B is the set of best matches of x with color s. □

Algorithm 7 therefore works correctly at least under idealized assumptions. (A0) is satisfied by construction for additive distance data. In real-life applications it is often possible to obtain at least a very good approximation.

Condition (A1) can be enforced by setting $\mathcal{Y} = L[s]$. This may be too expensive for large gene families and the inclusion of very distant relatives may be problematic for the construction of good multiple sequence alignments and thus, interfere with assumption (A2). In practice, it will therefore be necessary to limit $\mathcal{Y}$ to a manageable size and sufficient sequence similarity. In `ProteinOrtho` [144], for example, $\mathcal{Y} \subseteq L[s]$ is defined as the set of sequences with `blast` bit scores exceeding a certain fraction of the best hit for x in species s.

Condition (A2), i.e., the knowledge of appropriate outgroups, is more problematic. As discussed above, distance-based methods by construction do not convey information on the root of the phylogenetic tree T but only determine its unrooted version $\overline{T}$. As a consequence, additional information that is not contained in the pairwise distance measurements, is necessary to determine the edge in $\overline{T}$ that harbors the position of the root ρ of T [182]. In general, $\mathcal{Z}$ will be chosen from one or more species that are outgroups to the species X (containing x) and s in S. Even if outgroup species are given, gene duplications may predate the divergence of the available species set so that a given data set will usually violate (A2) for some pairs of leaves. We will return to this point in the next section.

7.4 IDENTIFICATION OF OUTGROUPS

In many practical applications, the phylogenetic relationships between the *species* under consideration are known. Similarly to the gene tree T, we model the species tree S as a planted tree with leaf set $L(S)$, where 0_S is the planted root of S with its only child ρ_S. As we have seen above, the most difficult issues arise when an outgroup is not readily available. In case that the root of the species tree S is known, however, we can still obtain some useful information. The main result of this section shows that inconsistency between gene and species quartets can be used to discard unsuitable outgroups.

In order to make use of the information in S, we need to describe the embedding of the planted gene tree T into S by the reconciliation map $\mu : V(T) \to V(S) \cup E(S)$, which, if restricted to duplication/loss scenarions, satisfies the axioms (R0), (R1), (R2), and (R3) (cf. Def. 6.1). Such reconciliation maps satisfy

$$\mu(x) \succeq_S \mathrm{lca}_S(\sigma(L(T(x)))), \tag{26}$$

i.e., an event $x \in V(T)$ in the gene tree cannot be mapped to a node in the species tree below the last common ancestor of all the species.

Ideally, the genes chosen as outgroup $\mathcal{Z}$ are co-orthologs, i.e., the duplication event that produced y' and y'' occured after the speciation event that separates the species Z from the two species X and Y.

Definition 7.2. *A duplication event $v \in V^0(T)$ in a gene tree T is called* ancient *if v is mapped to the edge $0_S\rho_S$ in the species tree S under the reconciliation map μ.*

The definition also applies to the subtree $S(u)$ rooted at an inner vertex $u \in V^0(S)$ of the species tree. It therefore makes sense to talk about duplication events that are ancient w.r.t. a given speciation event.

We will show in the following that inconsistency of gene and species quartets implies the existence of ancient duplications. To this end, we need some basic properties of reconciliation maps.

Lemma 7.3. *Let (T, σ) be a binary gene tree, S a species tree, and $\mu : V(T) \to V(S) \cup E(S)$ a reconciliation map without horizontal gene transfer. Let $x, y \in$*

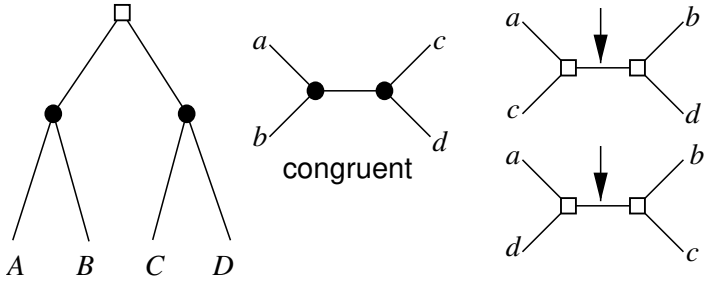


Fig. 42. Incongruence of gene and species quartets implies the existence of an ancient duplication. Consider four pairwise distinct species A, B, C, and D whose species tree is given on the l.h.s., and let four genes a, b, c, and d be chosen such that $\sigma(a) = A$, $\sigma(b) = B$, $\sigma(c) = C$, and $\sigma(d) = D$. The two speciation events separating A from B and C from D are indicated by ●. The root of this tree is indicated by □. One of the three possible gene quartets is congruent with the species tree, the other two are incongruent. For each of these quartets, Equ. (26) implies that the two inner vertices in these quartets cannot be mapped to the species tree below the root. The root of the gene tree must thus be mapped above the root of the species tree.

$L(T)$ be two genes with $\sigma(x) \neq \sigma(y)$. If $\mathrm{lca}_S(\sigma(x), \sigma(y)) \prec_S \mu(\mathrm{lca}_T(x, y))$, then $\mathrm{lca}_T(x, y)$ is a duplication event.

Proof. Assume, for contradiction, that $u := \mathrm{lca}_T(x, y)$ is a speciation event, i.e., $\mu(u) \in V^0(S)$. Let v' and v'' be the two children of u in T. Observe that $u := \mathrm{lca}_T(x, y)$ implies $x \in L(T(v'))$ and $y \in L(T(v''))$, or *vice versa*. W.l.o.g. assume $x \in L(T(v'))$ and $y \in L(T(v''))$. By (R3.i) and (R3.ii), $\mu(u) = \mathrm{lca}_S(\mu(v'), \mu(v''))$ and, in particular, $\mu(v')$ and $\mu(v'')$ are incomparable in S. Then, by Lemma 6.2, we have $\sigma(L(T(v'))) \cap \sigma(L(T(v''))) = \emptyset$. This and (R2) implies $\mu(v') \succeq_S \sigma(x)$ and $\mu(v') \succeq_S \sigma(y)$. The latter two arguments imply $\mathrm{lca}_S(\sigma(x), \sigma(y)) = \mu(u)$; a contradiction. □

The assumption that T is binary is necessary here as the example in Fig. 43 shows. Such reconciliations, however, cannot be meaningfully interpreted in terms of evolutionary events. Instead, the root of T confounds the duplication leading to x and y with the speciation separating $\mathrm{lca}_S(\sigma(x), \sigma(y))$ from $\sigma(z)$. To suppress such undesirable cases, we additionally require that μ satisfies:

(R5) If $\mu(\mathrm{lca}_T(x, y)) = \mu(\mathrm{lca}_T(x, z)) \in V^0(S)$, then $\mathrm{lca}_S(\sigma(x), \sigma(y)) = \mathrm{lca}_S(\sigma(x), \sigma(z))$.

In essence, (R5) ensures that a single node in T cannot represent two distinct speciation events, i.e., that the gene tree T is not "less resolved" than the species tree S into which it is embedded.

Lemma 7.4. *Let (T, σ) be a gene tree, S a species tree, and $\mu : V(T) \to V(S) \cup E(S)$ be a reconciliation map without horizontal gene transfer that satisfies* (R5). *Moreover, let $x, y \in L(T)$ be two genes with $\sigma(x) \neq \sigma(y)$. If $\mathrm{lca}_S(\sigma(x), \sigma(y)) \prec_S \mu(\mathrm{lca}_T(x, y))$, then $\mathrm{lca}_T(x, y)$ is a duplication event.*

Proof. We assume that T is non-binary since the binary case is covered already by Lemma 7.3. Moreover, we assume, for contradiction, that $u := \mathrm{lca}_T(x, y)$

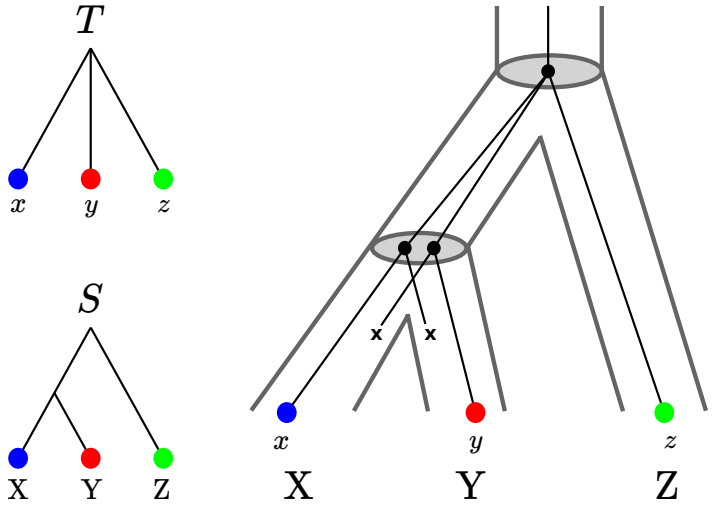


Fig. 43. A gene tree T, a species tree S, and their reconciliation. The reconciliation μ with $\mu(\mathrm{lca}_T(x,y,z)) = \mathrm{lca}_S(X,Y,Z)$, where $X = \sigma(x)$, $Y = \sigma(y)$, and $Z = \sigma(z)$, satisfies (R1), (R2), (R3.i), and (R3.ii) but does not admit an unambiguous interpretation of $\mathrm{lca}_T(x,y,z)$ as a single event: it confounds the speciation separating Z and $\mathrm{lca}_S(Y,X)$ with a gene duplication leading the ancestor of x and y or with the speciation separating X and Y. In either interpretation, the reconciliation map μ does not correspond to a mechanistic explanation of the gene family history.

is a speciation event, i.e., $\mu(u) \in V^0(S)$. Let v_x and v_y be the children of u with $x \preceq_T v_x$ and $y \preceq_T v_y$; thus we have $\sigma(x) \in \sigma(L(T(v_x)))$ and $\sigma(y) \in \sigma(L(T(v_y)))$. Since $u = \mathrm{lca}_T(x,y)$, v_x and v_y are incomparable in T and hence, $v_x \neq v_y$. By (R3.i), $\mu(v_x)$ and $\mu(v_y)$ are incomparable in S. Lemma 6.2 implies $\sigma(L(T(v'))) \cap \sigma(L(T(v''))) = \emptyset$ for all distinct children v' and v'' of u. The latter two facts together with (R2) imply $\mathrm{lca}_S(\sigma(x),\sigma(y)) = \mathrm{lca}_S(\mu(v_x),\mu(v_y)) \prec_S \mu(u)$. By (R3.i), $\mu(u) = \mathrm{lca}_S(\mu(v'),\mu(v''))$ for some children v' and v'' of u and thus, $\mathrm{lca}_S(\mu(v'),\mu(v'')) = \mathrm{lca}_S(\sigma(z'),\sigma(z''))$ for some leaves $z' \in L(T(v'))$ and $z'' \in L(T(v''))$ from different species $\sigma(z') \neq \sigma(z'')$.

We proceed by showing that for at least one of the species $\sigma(z')$ and $\sigma(z'')$ we have $\mathrm{lca}_S(\sigma(x),\sigma(z')) = \mathrm{lca}_S(\sigma(z'),\sigma(z''))$ or $\mathrm{lca}_S(\sigma(x),\sigma(z'')) = \mathrm{lca}_S(\sigma(z'),\sigma(z''))$. We suppose $\mathrm{lca}_S(\sigma(x),\sigma(z')) \neq \mathrm{lca}_S(\sigma(z'),\sigma(z''))$. Hence, we have $\mathrm{lca}_S(\sigma(x),\sigma(z')) \prec_S \mathrm{lca}_S(\sigma(z'),\sigma(z'')) = \mu(u)$ and therefore, $\mathrm{lca}_S(\sigma(x),\sigma(z'')) = \mathrm{lca}_S(\sigma(z'),\sigma(z''))$. Similarily, if $\mathrm{lca}_S(\sigma(x),\sigma(z'')) \neq \mathrm{lca}_S(\sigma(z'),\sigma(z''))$, then $\mathrm{lca}_S(\sigma(x),\sigma(z')) = \mathrm{lca}_S(\sigma(z'),\sigma(z''))$. Hence, assume w.l.o.g. $\mathrm{lca}_S(\sigma(x),\sigma(z')) = \mathrm{lca}_S(\sigma(z'),\sigma(z'')) \neq \mathrm{lca}_S(\sigma(x),\sigma(y))$. Now, by contraposition of (R5), we have $\mu(u) = \mu(\mathrm{lca}_T(x,y)) \neq \mu(\mathrm{lca}_T(x,z')) = \mu(u)$; a contradiction. □

Lemma 7.4 conveniently generalizes to sets of genes:

Corollary 7.1. *Let (T,σ) be a gene tree, S a species tree, and $\mu : V(T) \to V(S) \cup E(S)$ be a reconciliation map without horizontal gene transfer that satisfies* (R5) *and let $A \subseteq L(T)$ with $|\sigma(A)| \geq 2$. If $\mathrm{lca}_S(\sigma(A)) \prec_S \mu(\mathrm{lca}_T(A))$, then $\mathrm{lca}_T(A)$ is a duplication event.*

Proof. Note that $\mathrm{lca}_T(A) = \mathrm{lca}_T(x,y)$ holds for some $x,y \in A$. Assume first $\sigma(x) \neq \sigma(y)$. Thus $\mathrm{lca}_S(\sigma(A)) \prec_S \mu(\mathrm{lca}_T(A))$ implies $\mathrm{lca}_S(\sigma(x),\sigma(y)) \preceq_S$

$\mathrm{lca}_S(\sigma(A)) \prec_S \mu(\mathrm{lca}_T(A)) = \mu(\mathrm{lca}_T(x,y))$. Hence, the statement follows from Lemma 7.4. If $\sigma(x) = \sigma(y)$, then $\mathrm{lca}_T(A) = \mathrm{lca}_T(x,y)$ implies that there exist two distinct children v_x and v_y of $\mathrm{lca}_T(A)$ with $v_x \succeq x$ and $v_y \succeq y$. Thus $\mathrm{lca}_T(A) = \mathrm{lca}_T(v_x, v_y)$. However, since $\sigma(x) = \sigma(y)$ we have $\sigma(L(T(v_x))) \cap \sigma(L(T(v_y))) \neq \emptyset$. Thus Lemma 6.2 implies $\mu(\mathrm{lca}_T(A)) \notin V^0(S)$ and hence, $\mathrm{lca}_T(A)$ is duplication. □

We are now in the position to state the main result about inconsistent gene and species quartets. Consider four genes a, b, c, d residing in four pairwise distinct species A, B, C, and D, and assume that these four species form the quartet $(AB|CD)$. Then we say that the gene and species quartets are *congruent* if $\overline{T}[a,b,c,d] = (ab|cd)$ or $\times$. Otherwise, i.e., for $\overline{T}[a,b,c,d] \in \{(ac|bd), (ad|bc)\}$, they are called *incongruent*, see Fig. 42. In the following we show that the incongruence of gene and species quartets implies ancient duplications. More precisely:

Theorem 7.1. *Let A, B, C, and D be pairwise distinct species, set $u := \mathrm{lca}_S(A,B,C,D)$, $v_1 := \mathrm{lca}_S(A,B)$, and $v_2 := \mathrm{lca}_S(C,D)$. If $v_1 \prec_S u$, $v_2 \prec_S u$, and $\overline{T}[a,b,c,d] = (ac|bd)$ or $\overline{T}[a,b,c,d] = (ad|bc)$ for $a \in A$, $b \in B$, $c \in C$, $d \in D$, then $u \prec_S \mu(\mathrm{lca}_T(a,b,c,d))$ for every reconciliation map $\mu : V(T) \to V(S) \cup E(S)$ without HGT events. In particular, $\mathrm{lca}_T(a,b,c,d)$ is a duplication event.*

Proof. By assumption, $S_{|\{A,B,C,D\}}$ has the topology shown in Fig. 42. Assuming $(ac|bd)$, Equ. (26) implies $\mu(\mathrm{lca}_T(a,c)) \succeq_S \mathrm{lca}_S(\sigma(a), \sigma(c)) = u$ and $\mu(\mathrm{lca}_T(b,d)) \succeq_S \mathrm{lca}_S(\sigma(a), \sigma(c)) = u$. Thus both inner nodes p and q of the quartet are mapped no lower than u. The edge between them, therefore, must be mapped to an edge predating u since the speciation constraint (R3) implies that two $\prec_T$-comparable events in T, of which one is a speciation, cannot by mapped to the same vertex of S. Thus $u \prec_S \mu(\mathrm{lca}_T(a,b,c,d))$. The case $(ad|bc)$ is handled by an analogous argument exchanging c and d. The fact that $\mathrm{lca}_T(a,b,c,d)$ is a duplication event now follows from Lemma 7.4. □

This theorem can be used to discard suspicious outgroups: If $T[x,y,z_1,z_2]$ is incongruent with the known species tree, then $\sigma(z_1) \neq \sigma(z_2)$ should be replaced by outgroup candidates from earlier-branching species. The downside of using Thm. 7.1 is that it requires a systematic investigation of possibly large numbers of quartets.

In cases without too many ancient duplications we can use the following result about the inference of correct best matches from quartets. Recall that we assume that there are no HGT events. As it will be addressed in a forthcoming Master's thesis, the proof of the following result is omitted here. However, for the sake of completeness, the result is stated in this context:

Lemma 7.5. *Let (T,t,σ) be an event-labeled gene tree with $L = X \dot\cup Y \dot\cup Z$ and let S be the corresponding species tree on $S = \{X,Y,Z\}$ such that $\mathrm{lca}_S(X,Y) \prec \mathrm{lca}_S(X,Y,Z) = \rho_S$. Let μ be a reconciliation map for (T,t,σ) and S such that $|\mu^{-1}(\rho_S)| \leq 2$, and assume (A0). Then Algorithm 7, using $\mathcal{Y} \subseteq Y$ as the candidate best match set and $\mathcal{Z} \subseteq Z$ as outgroup set, correctly determines, for every gene $x \in X$, all best matches in species Y.*

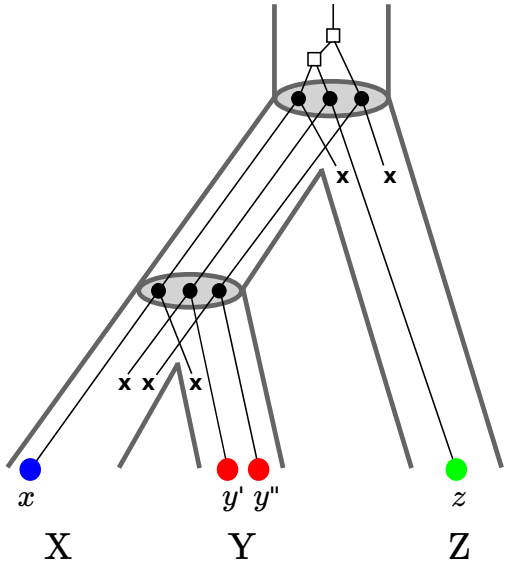


Fig. 44. A reconciliation scenario with two ancient duplications leading to false positive and false negative best matches of the gene x. Assuming that y', y'' are the only extant genes of species Y and z is an outgroup for $\{x, y', y''\}$, Algorithm 7 incorrectly infers the quartet $(xy''|y'z)$, i.e., y'' is inferred as the unique best match of x in species Y. In fact, (x, y'') corresponds to a false positive, while (x, y') is a false negative edge in the inferred best match graph.

The condition $|\mu^{-1}(\rho_S)| \leq 2$ makes an explicit assumption on the true history of the gene family by limiting the scenario to at most one ancient duplication on $X \cup Y \cup Z$. Fig. 44 shows that this condition cannot be dropped: if there are two or more ancient duplications affecting X, Y, and Z, then the correct inference of best matches from quartets can no longer be guaranteed.

It seems likely that it is possible in most cases to unambiguously identify pairs whose last common ancestor in the gene tree predates the last common ancestor of the species tree under consideration. While it may be difficult to determine the relative order of such duplications, it appears likely that clustering methods used to extract groups of co-orthologs can be adapted to disentangle such ancient "paralog groups".

7.5 SUMMARY

The idea to use quartet structures for improvement of orthology estimates is not new, see e.g. [236] or the use of quartets as witnesses of non-orthology in OMA [225]. This chapter investigated in detail how and when quartets can help to improve and/or correct empirical best hit data to identify best matches in the sense of closest evolutionary relatives. A workflow was presented that, given an additive distance among the genes, correctly identifies best matches under ideal conditions, which seem likely to be approximated in real applications. Moreover, we observed that only local, qualitative information is necessary and, in particular, there is no need to attempt a detailed reconstruction of the rooted or unrooted gene trees. Instead, it suffices to operate on a moderate subset of quartets. The key observation is that this, however, crucially depends on the ability to identify reliable outgroup genes from a third species. First simulation results that evaluate how well different approaches – including the workflow presented here – estimate best matches (in the sense of evolutionary relatedness) from both perfect and noisy data, can be found in [211].

8

RECONSTRUCTING GENE TREES FROM FITCH'S XENOLOGY RELATION

While best match heuristics have been very successful as approximations of the orthology relation [10, 169], no comparable approach to extract the xenology relation directly from (dis)similarity data has been devised to-date. Presumably this is at least one reason why the binary xenology relation has attracted very little attention so far. However, as we have seen in Chapter 6, even a relatively small amount of HGT has a major impact in terms of missing orthology edges in the reciprocal best match graph, thus understanding the xenology relation is crucial not only for detecting HGT but also for correct orthology assignment.

The main focus of this part lies on the mathematical properties of the non-symmetric xenology relation $\mathcal{X}$. In particular, we will be concerned with two related questions: (1) How much information on the gene tree T and the location of the horizontal transfer events within T is contained in the xenology relation? (2) Is it possible to extract the topological information and labeling information from $\mathcal{X}$ efficiently?

This chapter shows that valid non-symmetric xenology relations correspond to a heritable family of digraphs, the so-called Fitch graphs. These are characterized by a small set of forbidden subgraphs on three vertices and thus can be recognized in cubic time (Section 8.3). Fitch graphs form a subclass of di-cographs, which have recently been associated with an alternative concept of xenology [100]. Each Fitch graph is explained by a unique least resolved edge-labeled phylogenetic tree which is displayed by the full evolutionary scenario (Section 8.2). It therefore provides at least partial information on the gene tree and the placement of the horizontal transfer events. It will be demonstrated, furthermore, that this tree as well as the corresponding edge labeling can be constructed from $\mathcal{X}$ in polynomial time (Section 8.4). Features of heritable graph properties lead to a linear-time recognition algorithm, as well as NP-completeness and fixed-parameter tractable results for the respective graph modification problems. Finally, in Section 8.5, the xenology relation will be extended to the symmetric Fitch relation by considering the undirected version of $\mathcal{X}$, which turns out to correspond to complete multipartite graphs. We start this chapter with a formal definition and some simple results about the non-symmetric Fitch relation.

The results of this chapter have been published in Geiß et al. [72] and Hellmuth et al. [101].

8.1 THE DIRECTED (FITCH-)XENOLOGY RELATION

In this chapter, we are interested in rooted phylogenetic trees $T = (V, E)$ with leaf set $L = L(T)$ that are endowed with edge labels $\lambda : E \to \{0, 1\}$ such that

$$\lambda(e) = \begin{cases} 1 & \text{if } e \text{ is a horizontal transfer edge} \\ 0 & \text{otherwise} \end{cases}$$

For simplicity we will speak of 0-edges and 1-edges in T depending on their labeling. Edge-labeled trees will be written as (T, λ). Unless explicitly stated otherwise, all trees in this chapter are assumed to be rooted. The first part of the chapter is concerned with the following, directed relation:

Definition 8.1. *Given an edge-labeled phylogenetic tree (T, λ) with leaf set L we set $(x, y) \in \mathcal{X}_{(T,\lambda)}$ for $x, y \in L$ whenever there is at least one directed horizontal transfer event between y and the last common ancestor of x and y, i.e., if the uniquely defined path from $\mathrm{lca}_T(x, y)$ to y contains at least one 1-edge. We write $[x, y] \in \mathcal{X}_{(T,\lambda)}$ if both edges (x, y) and (y, x) are contained in $\mathcal{X}_{(T,\lambda)}$ and $x|y$ if $\mathcal{X}_{(T,\lambda)}$ contains neither (x, y) nor (y, x).*

By construction $\mathcal{X}_{(T,\lambda)}$ is irreflexive; hence it can be regarded as a simple directed graph. Similarly to BMGs and RBMGs, we will therefore interchangeably speak of $\mathcal{X}_{(T,\lambda)}$ as graph or relation. It is easy to check that $\mathcal{X}_{(T,\lambda)}$ is in general neither symmetric nor antisymmetric. The relation $\mathcal{X}_{(T,\lambda)}$ formalizes Fitch's concept of *xenology* [66] (see also Chapter 3).

Definition 8.2. *An edge-labeled phylogenetic tree (T, λ)* explains *a given irreflexive relation $\mathcal{X}$ whenever $\mathcal{X} = \mathcal{X}_{(T,\lambda)}$.*
A relation $\mathcal{X}$ is valid *if there exists an edge-labeled tree that explains $\mathcal{X}$, and* invalid *otherwise.*

Hence, an edge-labeled tree (T, λ) explains a relation $\mathcal{X}$ if there is a 1-edge on the path from $\mathrm{lca}(x, y)$ to y if and only if $(x, y) \in \mathcal{X}$. By construction, $\mathcal{X}$ must be defined on $L(T)$. However, we will sometimes abuse notation and say that $\mathcal{X}[L']$ is explained by (T, λ) for some $L' \subseteq L$ if $(T_{|L'}, \lambda_{|L'})$ explains $\mathcal{X}[L']$. An example of a gene tree with the corresponding Fitch relation $\mathcal{X}$ and an edge-labeled tree that explains $\mathcal{X}$ can be found in Fig. 45.

The notion of a tree T' being displayed by a tree T can be generalized to edge-labeled trees: We say that (T', λ') is *displayed* by (T, λ) if T' is displayed by T in the usual sense and an edge $e' \in E(T')$ has label $\lambda'(e') = 1$ if and only if the path in T that corresponds to e' contains at least one 1-edge.

Lemma 8.1. *Let (T', λ') be a tree with leaf set $L' = L(T')$ that is displayed by (T, λ). Then $\mathcal{X}_{(T',\lambda')}$ is the subgraph of $\mathcal{X}_{(T,\lambda)}$ induced by L'.*

Proof. Consider two distinct leaves $x, y \in L'$. By construction of (T', λ'), there is a 1-edge on the path from $\mathrm{lca}_{T'}(x, y)$ to the leaf y in (T', λ') if and only if the corresponding path in (T, λ) contains a 1-edge and thus, $(x, y) \in \mathcal{X}_{(T',\lambda')}$ if and only if $(x, y) \in \mathcal{X}_{(T,\lambda)}$. □

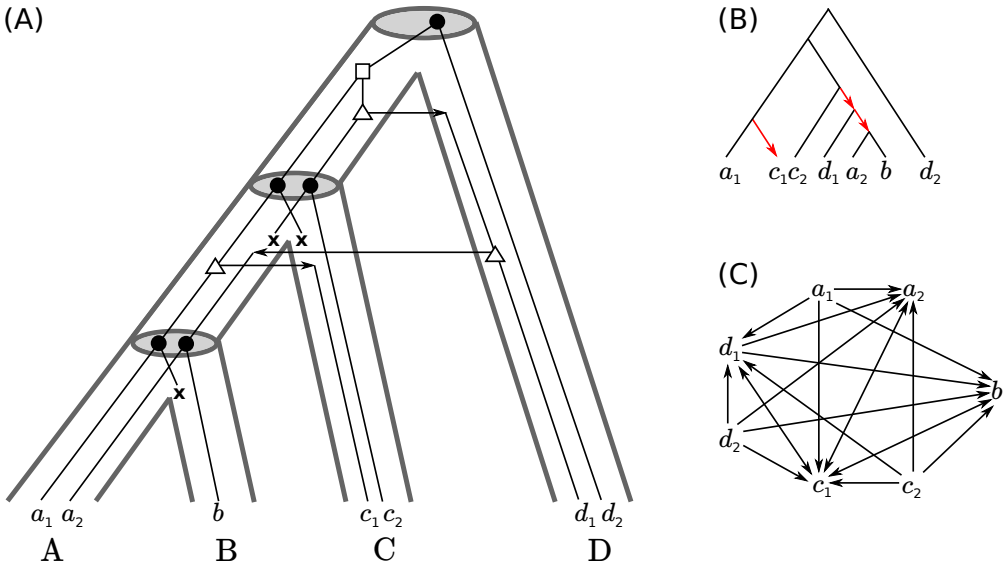


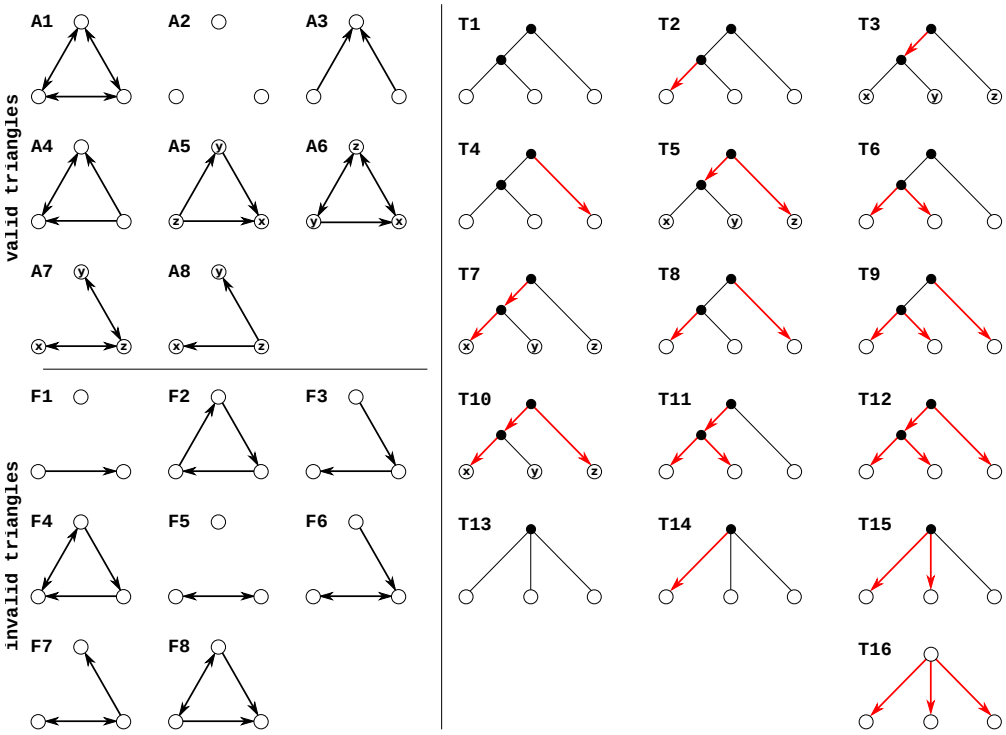
Fig. 45. (A) Event-labeled gene tree embedded in the (tube-like) species tree. The leaf set of the gene tree are the genes a_1, a_2, b, c_1, c_2, d_1, and d_2 in the genomes of the four species A, B, C, and D. The gene tree contains speciations (●), duplications (□), HGT events (△) and gene losses (×). (B) Removal of all gene losses, suppression of all resulting vertices of degree 2, and ignoring the types of the events on the inner vertices yields an edge-labeled tree in which the transfer edges are labeled by 1 (red arrow) and all other edges by 0 (black edges). Panel (C) shows the Fitch graph explained by the edge-labeled tree of Panel (B).

The enumeration of all edge-labeled trees on two vertices shows that all four possible digraphs on two vertices are valid. For three vertices, however, there are valid and invalid digraphs. These are summarized in Figure 46: up to isomorphism there are eight valid A_1-A_8 and eight invalid F_1-F_8 digraphs. We will refer to them as valid and invalid *triangles*. An enumerative approach to find all valid and invalid triangles has been developed by Anders [12] in the context of his master thesis.
As we shall see in Section 8.3, a relation $\mathcal{X}$ is valid, i.e., it has a tree representation, if and only if all its triangles are valid. This gives rise to the following definition:

Definition 8.3. *An irreflexive binary relation $\mathcal{X}$ on L is a* Fitch relation *if all its triangles are valid. Its graph representation is called a* Fitch graph.

A graph G is a di-cograph if and only if it does not contain one of the digraphs shown in Fig. 47 as an induced subgraph [40]. Since each of these graphs contains one of the forbidden triangles, every Fitch graph is also a di-cograph. On the other hand, a di-cograph that does not contain F_1, F_5, or F_8 as an induced subgraph is a Fitch graph. As an immediate consequence of its characterization in terms of forbidden induced subgraphs, Fitch graphs are a heritable family, i.e., every induced subgraph of a Fitch graph is again a Fitch graph. We summarize these observations for later reference as

Lemma 8.2. *The Fitch graphs are a heritable subfamily of the di-cographs.*



A1	A2	A3	A4	A5	A6	A7	A8
T9, T12, T16	T1, T13	T2, T4, T14	T6, T8, T11, T15	T7	T10	T5	T3

Fig. 46. *Upper Left:* Shown is the graph representation for all possible relations $\mathcal{X} \subseteq L \times L$ with $|L| = 3$. The relations are grouped into valid (A_1-A_8) and non-valid (F_1-F_8).
Upper Right: All possible (up to isomorphism) subtrees on three leaves of a tree (T, λ) are shown. Edges can be understood as paths, whereby red (resp. black) edges indicate that there is (resp., is not) a 1-edge on the particular path.
Lower Part: The table shows which tree explains which relation. In particular, there is no tree that explains one of the graphs F_1 to F_8.

A closer inspection shows that four of the eight valid triangles, namely A_1-A_4 can be explained by multiple trees, including one of the non-binary trees T_{13} to T_{16}. In contrast, each of the triangles A_5-A_8 with a given labeling of its three leaves is explained by a unique edge-labeled binary tree, i.e., a specific labeled triple.

Definition 8.4. *An edge-labeled triple $ab|c$ is* informative *if it explains a labeled triangle isomorphic to one of A_5, A_6, A_7, or A_8.*

Thus, if $\mathcal{X}$ contains a triangle of the form A_5, A_6, A_7, or A_8 as an induced subgraph, then any tree explaining $\mathcal{X}$ must display the corresponding informative triple. Any valid relation $\mathcal{X}$ can therefore be associated with a uniquely defined set $\mathcal{R}(\mathcal{X})$ of *informative triples* that it displays: $r \in \mathcal{R}(\mathcal{X})$ if and only if r is the unique edge-labeled triple explaining an induced triangle isomorphic to A_5, A_6, A_7, or A_8. For later reference we summarize this fact as

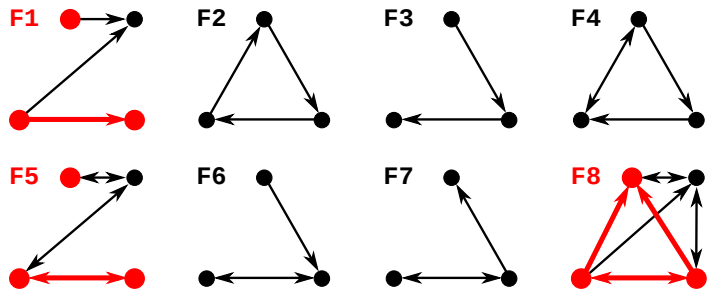


Fig. 47. The eight digraphs are the forbidden induced subgraphs that characterize di-cographs [60, 40]. The five digraphs on three vertices correspond to five of the eight forbidden triangles. Each digraph on four vertices contains one of the remaining forbidden triangles (highlighted by bold-red edges and vertices).

Lemma 8.3. *If (T, λ) explains $\mathcal{X}$, then all triples in $\mathcal{R}(\mathcal{X})$ must be displayed by (T, λ).*

8.2 LEAST RESOLVED EDGE-LABELED PHYLOGENETIC TREES

In general, there may be more than one rooted phylogenetic tree that explains a given relation $\mathcal{X}$. In particular, if $\mathcal{X}$ is explained by a non-binary tree (T, λ), then there is always a binary tree (T', λ') that refines T and explains the same relation $\mathcal{X}$ by setting $\lambda'(e) = \lambda(e)$ for all edges e that are also in T and by choosing the label $\lambda'(e) = 0$ for all edges e that are not contained in T. In this section, we will see that whenever a relation $\mathcal{X}$ is explained by an edge-labeled tree (T, λ), then there exists a unique "smallest" tree with this property, which we will call the least resolved tree. These least resolved trees will play a key role for obtaining a characterization of Fitch relations in the following.

Definition 8.5. *Let $(T = (V, E), \lambda)$ be an edge-labeled phylogenetic tree and let $e = uv \in E$. The phylogenetic tree (T^e, λ^e), referred to as the* extended contraction *of e in (T, λ), is obtained by the following procedure:*
First contract the edge e in T and keep the edge labels of all non-contracted edges. If e is an inner edge, the resulting tree is again a phylogenetic tree and we are done. The contraction of an outer edge $e = uv$, however, leads to (i) the loss *of a leaf v and (ii) a decrease in the degree of the parental vertex u. The latter may violate the degree conditions required for a phylogenetic tree. If u is the root of T that has degree 1 in T^e, we delete u and its incident edge, and declare the unique remaining child of u as the root of T^e. Thus T^e is obtained by an additional (simple) contraction of the edge $\rho_{T^e}\mathsf{child}_{T^e}(\rho_{T^e})$. Otherwise, if u is an inner vertex that has degree 2 after the contraction of e, we apply an additional (simple) contraction of the edge $u\mathsf{child}_{T^e}(u)$ and set $\lambda(\mathsf{par}_{T^e}(u)u) = 1$ if $\lambda(u\mathsf{child}_{T^e}(u)) = 1$. Equivalently, the path from the parent w of u to the unique remaining child w' of u is replaced by a single edge ww'. This edge is a 1-edge if and only if at least one of the edges wu and uw' in the initial tree was a 1-edge.*
The edge e is said to be ex-*contracted in (T, σ).*

In order to avoid confusion with extended edge contractions, we will often refer to contractions as defined in Chapter 3 as *simple contractions*. Note in particular that extended contraction of an inner edge is equivalent to a simple contraction.

Definition 8.6. *An edge-labeled phylogenetic tree* $(T = (V, E), \lambda)$ *is* least resolved (w.r.t. $\mathcal{X}_{(T,\lambda)}$) *if none of the trees* (T^e, λ^e) *obtained from* (T, σ) *by an extended contraction of* $e \in E$*, explains* $\mathcal{X}_{(T,\lambda)}$*.*

The notion of "least resolved trees" in this chapter always refers to xenology relations. For simplicity of notation, we will therefore drop the explicit reference to the corresponding relation whenever the context is clear.
It is easy to see that (T^e, λ^e) is, by construction, always obtained by a sequence of simple edge contractions and thus, (T^e, λ^e) is displayed by (T, λ).
We state here the main result of this section:

Theorem 8.1. *Let* $\mathcal{X} \subseteq L \times L$ *be a valid relation,* (T, λ) *a phylogenetic tree that explains* $\mathcal{X}$ *and let* $(\widehat{T}, \widehat{\lambda})$ *be a least resolved phylogenetic tree w.r.t.* $\mathcal{X}$*. Then* (T, λ) *displays* $(\widehat{T}, \widehat{\lambda})$*. Moreover, the tree* $(\widehat{T}, \widehat{\lambda})$ *has the minimum number of vertices among all trees that explain* $\mathcal{X}$*, and is unique.*

In order to prove Thm. 8.1, we need the following intermediate results, Lemma 8.4 - 8.11.

Lemma 8.4. *Let* (T, λ) *be an edge-labeled phylogenetic tree. If* e *is an inner 0-edge in* (T, λ)*, then* $\mathcal{X}_{(T^e,\lambda^e)} = \mathcal{X}_{(T,\lambda)}$*. If* e *is an inner 1-edge, then* $\mathcal{X}_{(T^e,\lambda^e)} \subseteq \mathcal{X}_{(T,\lambda)}$*.*

Proof. The (extended) contraction of the inner 0-edge $e = uv$ does not change the number of 1-edges along the paths connecting any two leaves. It affects the last common ancestor of x and y if $\mathrm{lca}_T(x, y) = u$ or $\mathrm{lca}_T(x, y) = v$. In either case, however, the number of 1-edges between $\mathrm{lca}_T(x, y)$ and the leaves x and y remains unchanged. Hence, the relation $\mathcal{X}_{(T,\lambda)}$ is not affected by the contraction.

The (extended) contraction of an inner 1-edge e reduces the number of 1-edges along the path between all pairs of leaves whose connecting path in T contain e. Thus, if $(x, y) \in \mathcal{X}_{(T^e,\lambda^e)}$, then the path connecting x and y in T contains also at least one 1-edge and hence, $(x, y) \in \mathcal{X}_{(T,\lambda)}$. □

Note that edge contractions therefore always imply $\mathcal{X}_{(T^e,\lambda^e)} \subseteq \mathcal{X}_{(T,\lambda)}$. An example for $\mathcal{X}_{(T^e,\lambda^e)} \subset \mathcal{X}_{(T,\lambda)}$ is given by the tree T_3 in Fig. 46 and contraction of the single inner 1-edge.
There may be edges in a tree whose labeling does not affect the relation, i.e., they can be labeled either 0 or 1. This observation gives rise to the following definition:

Definition 8.7. *An edge* e *in a tree* (T, λ) *is* irrelevant *if* (T, λ') *with* $\lambda'(e) \neq \lambda(e)$ *and* $\lambda'(f) = \lambda(f)$ *for all* $f \neq e$ *still explains* $\mathcal{X}_{(T,\lambda)}$*. Edges that are not irrelevant are called* relevant*.*

As an example consider the two trees T_9 and T_{12} in Figure 46. Both explain the valid triangle A_1. The inner edge of T_9 and T_{12} is a 0-edge and 1-edge, respectively. Thus this edge is irrelevant. The tree T_{16}, which is obtained from both T_9 and T_{12} by contracting the irrelevant edge, still explains A_1. For later reference, we provide a simple characterization of irrelevant edges.

Lemma 8.5. *An edge $e = uv$ is irrelevant in a phylogenetic tree (T, λ) if and only if e is an inner edge and every path from v to each leaf in the subtree rooted at v contains a 1-edge.*

Proof. Any inner edge e that satisfies the condition of the lemma is irrelevant because every path from u to a leaf contains a 1-edge irrespective of the label of uv.

Conversely, assume first that $e = uv$ is an outer edge. Hence, changing the label of e would immediately change the relation between v and any leaf w located in a subtree rooted at a sibling of v. Since at least one such leaf w exists in a phylogenetic tree, e is relevant. Now suppose that $e = uv$ is an inner edge and that there is a leaf w below v such that the path from v to w comprises only 0-edges. Let x be a leaf such that $\mathrm{lca}(w, x) = u$. Since T is a phylogenetic tree, such a leaf always exists. Then $(x, w) \in \mathcal{X}$ if and only if $\lambda(e) = 1$, i.e., the inner edge e is relevant. □

A crucial consequence of Lemma 8.5 is that every outer edge is relevant. Furthermore, since an irrelevant edge can be relabeled as a 0-edge without affecting $\mathcal{X}_{(T,\lambda)}$, Lemma 8.4 implies that irrelevant edges can be (*ex*-)contracted without changing $\mathcal{X}_{(T,\lambda)}$. These observations naturally pose the question how edge-labeled trees are structured that cannot be contracted further without affecting $\mathcal{X}_{(T,\lambda)}$.

Lemma 8.6. *Let (T, λ) be an edge-labeled phylogenetic tree explaining $\mathcal{X}$. Then the tree (T^e, λ^e) obtained by extended contraction of the edge e explains $\mathcal{X}$ if and only if e is irrelevant or e is an inner 0-edge.*

Proof. The discussion above already shows that irrelevant edges as well as 0-edges can be *ex*-contracted without affecting $\mathcal{X}$. We show that $\mathcal{X}_{(T^e,\lambda^e)} \neq \mathcal{X}_{(T,\lambda)}$ whenever e is an outer edge or a relevant inner 1-edge. First we assume that e is an outer edge. Clearly, if v is a leaf, then extended contraction of $e = uv$ would change v to an inner vertex in (T^e, λ^e). Thus $L(T) \neq L(T')$ and therefore, (T^e, λ^e) does not explain $\mathcal{X}$. Now, let e be a relevant inner 1-edge. Then there is a leaf x in the subtree rooted at v such that the path from v to x consists only of 0-edges (cf. Lemma 8.5). Since (T, λ) is phylogenetic, there exists a leaf $y \in L(T)$ such that $\mathrm{lca}_T(x, y) = u$. Moreover, as $\lambda(uv) = 1$, we have $(y, x) \in \mathcal{X}$. Contracting e makes the vertex u^*, obtained by identifying u and v, the last common ancestor of x and y, i.e., $\mathrm{lca}_{T^e}(x, y) = u^*$. The path from u^* to x now contains only 0-edges, i.e., $(y, x) \notin \mathcal{X}_{(T^e,\lambda^e)}$. Thus, relevant 1-edges of (T, λ) cannot be *ex*-contracted without affecting $\mathcal{X}$. □

The following result shows that relevant edges in a tree (T, λ) remain relevant in any of its *ex*-contracted versions (T^e, λ^e), where e is an inner 0-edge or an irrelevant edge.

Lemma 8.7. *Let (T, λ) be an edge-labeled phylogenetic tree explaining $\mathcal{X}$, the edge e be an inner 0-edge or an irrelevant 1-edge in (T, λ), and (T^e, λ^e) the tree obtained from (T, λ) by extended contraction of e. Then, the edge $f \neq e$ is relevant in (T^e, λ^e) if and only if f is relevant in (T, λ).*

Proof. As a consequence of Lemma 8.6, (T^e, λ^e) still explains $\mathcal{X}$. Lemma 8.5 implies that the edge $f = uv$ is irrelevant in (T^e, λ^e) if and only if f is an inner edge and all paths from v to leaves below v contain a 1-edge. If e is not located below f, then the extended contraction of e does not affect this condition and thus, f is irrelevant in (T^e, λ^e) if and only if it is irrelevant in (T, λ).

Now suppose e is located below f. If e was a 0-edge, the number of 1-edges along the paths from v to the leaves does not change upon extended edge contraction and thus, f is irrelevant in (T^e, λ^e) if and only if it is irrelevant in (T, λ). Finally, suppose $e = u'v'$ was an irrelevant 1-edge. Thus we can set $\lambda(e) = 0$ in (T, λ) without changing the relation $\mathcal{X}$. Now we can repeat the latter arguments to conclude that f is irrelevant in (T^e, λ^e) if and only if it is irrelevant in (T, λ). □

The following result shows that the order of the extended contraction of inner 0-edges or irrelevant 1-edges does not affect the resulting relation.

Lemma 8.8. *Let (T, λ) be an edge-labeled phylogenetic tree and let e and f be two edges in T such that (T, λ), (T^e, λ^e), and (T^f, λ^f) explain the same relation $\mathcal{X}$. Then, (T^{ef}, λ^{ef}) obtained from (T^e, λ^e) by extended contraction of the edge f, also explains $\mathcal{X}$.*

Proof. By Lemma 8.6, an edge can be *ex*-contracted without affecting $\mathcal{X}$ if and only it is an inner 0-edge or an irrelevant 1-edge. The labeling of f is not affected by extended contraction of e and *vice versa*. Lemma 8.7 furthermore shows that the (ir)relevance of an edge $f \neq e$ is conserved by the extended contraction of 0-edges and irrelevant 1-edges. Therefore e and f can be *ex*-contracted in arbitrary order and preserve $\mathcal{X}$ in each contraction step. □

We will now apply the results developed so far to least resolved trees. First, we show that the order of extended edge contractions does not affect the resulting least resolved tree. In particular, the importance of the next lemma is given by the following observation: By definition, (T, λ) is least resolved w.r.t. $\mathcal{X}$ if none of its single edge contracted trees (T^e, λ^e) explains $\mathcal{X}$. However, this does not directly imply that there is no sequence of extended edge contractions that may yield a tree that explains $\mathcal{X}$.

Lemma 8.9. *Let (T, λ) be a least resolved tree w.r.t. $\mathcal{X} = \mathcal{X}_{(T,\lambda)}$. Then, there is no sequence of extended edge contractions $e_1 e_2 \dots e_\ell$ such that the resulting contracted tree $(T^{e_1 e_2 \dots e_\ell}, \lambda^{e_1 e_2 \dots e_\ell})$ explains $\mathcal{X}_{(T,\lambda)}$.*

Proof. Let (T, λ) be a least resolved tree, i.e., none of the *ex*-contracted trees (T^e, λ^e), $e \in E$, explains $\mathcal{X}_{(T,\lambda)}$. Lemma 8.4 and 8.6 imply that any edge $e \in E$ must be either an outer edge or a relevant 1-edge. Clearly, if one edge of the sequence $e_1 e_2 \dots e_\ell$ is an outer edge, then the statement is trivially satisfied.

Hence, assume that all edges $e_1 e_2 \dots e_\ell$ are inner edges and therefore, relevant 1-edges in (T, λ). Lemma 8.4 implies that for $\mathcal{X}$ to change, there must be at

least one pair of leaves x, y such that $(x, y) \in \mathcal{X}_{(T,\lambda)}$ and $(x, y) \notin \mathcal{X}_{(T^e,\lambda^e)}$, i.e., there is no 1-edge along the path from $\mathrm{lca}(x, y)$ to y in T^e, and e was the only 1-edge along the path from $\mathrm{lca}(x, y)$ to y in T. By Lemma 8.4, this implies $(x, y) \notin \mathcal{X}'$ for the relation explained by any tree that is obtained from extended edge contractions of (T^e, λ^e), i.e., there is no sequence of extended edge contractions that leads to a tree (T', λ') such that $\mathcal{X}_{(T',\lambda')} = \mathcal{X}_{(T,\lambda)}$. □

Next, we summarize some useful properties of least resolved trees that will be used repeatedly in the following sections.

Lemma 8.10. *Let (T, λ) be a phylogenetic tree that explains $\mathcal{X}$. The following three conditions are equivalent:*

1. *(T, λ) is least resolved tree w.r.t. $\mathcal{X}$.*
2. *Every edge of (T, λ) is relevant and all inner edges are 1-edges.*
3. *(a) Every inner edge of (T, λ) is a 1-edge.*
   (b) For every inner edge uv there is an outer 0-edge vx in (T, λ).

Moreover, if (T, λ) is least resolved w.r.t. $\mathcal{X}$, then

4. *Any inner edge of (T, λ) is distinguished by at least one informative rooted triple in $\mathcal{R}(\mathcal{X})$,*
5. *For any edge-contracted tree (T^e, λ^e) of (T, λ) there is a triple in $\mathcal{R}(\mathcal{X})$ that is not displayed by (T^e, λ^e), i.e., (T, λ) is also least resolved w.r.t. $\mathcal{R}(\mathcal{X})$, and*
6. *The tree $(T(v), \lambda_{|L(T(v))})$, that is the subtree of T rooted at the vertex v with $\lambda_{|L(T(v))}(e) = \lambda(e)$ for any edge e of $T(v)$, is least resolved w.r.t. the subrelation $\mathcal{X}[L(T(v))]$ of $\mathcal{X}$.*

Proof. The equivalence of Conditions 1 and 2 is an immediate consequence of Lemma 8.6. Moreover, by Lemma 8.4, Condition 1 implies Condition 3(a). To see that also Condition 3(b) is implied given Conditions 1 or 2, observe that if v is incident to 1-edges only, then Lemma 8.5 implies that uv is irrelevant. Thus v must be incident to at least one 0-edge. However, this 0-edge cannot be an inner edge because inner 0-edges can always be (*ex*-)contracted due to Lemma 8.4. Thus v is incident to an outer 0-edge.

Now assume that Condition 3 is satisfied. First observe that none of the outer edges can be *ex*-contracted without changing $\mathcal{X}$. Let uv be an inner 1-edge and vx an outer 0-edge. Since (T, λ) is phylogenetic, there is a leaf y for which $\mathrm{lca}(x, y) = u$. Thus $(y, x) \in \mathcal{X}$. However, extended contraction of the inner edge uv would yield $(y, x) \notin \mathcal{X}$. Thus none of the inner edges can be *ex*-contracted and therefore, (T, λ) is least resolved w.r.t. $\mathcal{X}$.

Property 4: Consider an arbitrary inner edge $e = uv$ of T. Since (T, λ) is phylogenetic, there are necessarily leaves x, y, and z such that $\mathrm{lca}(x, y) = v$ and $\mathrm{lca}(x, y, z) = u$. Since e is a 1-edge due to Property 3, the tree on $\{x, y, z\}$ displayed by T must be one of $T_3, T_5, T_7, T_{10}, T_{11}$, or T_{12} in Fig. 46, where the red inner edge denotes the edge e. One easily checks explicitly that neither

T_{11} nor T_{12} is least resolved since contraction of e still yields $\mathcal{X}_{(T^e,\lambda^e)} = \mathcal{X}_{(T,\lambda)}$. The remaining trees T_3, T_5, T_7, and T_{10}, on the other hand, are informative triples $xy|z \in \mathcal{R}(\mathcal{X})$. Since $\mathrm{lca}(x,y) = v$ and $\mathrm{lca}(x,y,z) = u$, the edge e is by definition distinguished by the triple in $xy|z \in \mathcal{R}(\mathcal{X})$.

Property 5: Recall from Property 4 that each inner edge $e = uv$ is distinguished by a triple $xy|z \in \mathcal{R}(\mathcal{X})$; therefore $\mathrm{lca}(x,y) = v$ and $\mathrm{lca}(x,y,z) = u$. However, extended contraction of e would yield $\mathrm{lca}_{T^e}(x,y) = \mathrm{lca}_{T^e}(x,y,z)$, which in turn would imply that $xy|z \in \mathcal{R}(\mathcal{X})$ is not displayed by (T^e, λ^e); a contradiction.

Property 6: By construction, no edge ab with $v \succeq_T a$ was removed in $T(v)$. Since $\lambda_{|L(T(v))}(e) = \lambda(e)$ for any edge e of $T(v)$, Property 3 is trivially fulfilled in $(T(v), \lambda_{|L(T(v))})$. Thus $(T(v), \lambda_{|L(T(v))})$ is least resolved w.r.t. $\mathcal{X}[L(T(v))]$. □

As an immediate consequence of Lemma 8.8, which implies that all extended edge contractions can be performed independently of each other, we observe that for every edge-labeled tree (T,λ) there exists a unique least resolved tree $(\widehat{T}, \widehat{\lambda})$ that can be obtained from (T,λ) by a sequence of extended edge contractions. Every tree explaining $\mathcal{X}$ is therefore a refinement of a least resolved tree that explains $\mathcal{X}$. By Lemma 8.3, any tree that explains $\mathcal{X}$ must display the triples in $\mathcal{R}(\mathcal{X})$. An even stronger result holds however:

Lemma 8.11. *If (T,λ) is a least resolved tree w.r.t. $\mathcal{X} = \mathcal{X}_{(T,\lambda)}$, then $\mathcal{R}(\mathcal{X})$ identifies (T,λ).*

Proof. If $\mathcal{R}(\mathcal{X}) = \emptyset$, then, by construction, all induced subgraphs on three vertices must be isomorphic to one of the graphs A_1, A_2, A_3, or A_4 in Fig. 46. In this case, (T,λ) is a star tree, i.e., an edge-labeled tree that consists of outer edges only. Otherwise, (T,λ) contains inner edges that are, by Lemma 8.10, distinguished by at least one informative rooted triple in $\mathcal{R}(\mathcal{X})$, contradicting that $\mathcal{R}(\mathcal{X}) = \emptyset$. Hence, $r(T) = \emptyset$, and therefore, $r(T) = \mathrm{cl}(\mathcal{R}(X))$. Lemma 3.1 implies that $\mathcal{R}(\mathcal{X})$ identifies (T,λ).

In the case $\mathcal{R}(\mathcal{X}) \neq \emptyset$, assume for contradiction that $r(T) \neq \mathrm{cl}(\mathcal{R}(\mathcal{X}))$. By Lemma 8.3, we have $\mathcal{R}(\mathcal{X}) \subseteq r(T)$. Isotony of the closure (see Thm. 3.1(3) in [25] and Section 3.3.5), ensures $\mathrm{cl}(\mathcal{R}(\mathcal{X})) \subseteq \mathrm{cl}(r(T)) = r(T)$. Our assumption therefore implies $\mathrm{cl}(\mathcal{R}(\mathcal{X})) \subsetneq r(T)$ and thus, the existence of a triple $ab|c \in r(T) \setminus \mathrm{cl}(\mathcal{R}(\mathcal{X}))$. In particular, therefore, $ab|c \notin \mathcal{R}(\mathcal{X})$. Note that neither $ac|b$ nor $bc|a$ can be contained in $\mathcal{R}(\mathcal{X})$ since (T,λ) explains $\mathcal{X}$ and, by assumption, already displays the triple $ab|c$. Thus $\mathcal{R}(\mathcal{X})$ contains no triples on $\{a,b,c\}$.

Lemma 8.10 implies that there exists a vertex $v \in \mathsf{child}(\mathrm{lca}(a,b,c))$, with $v \succeq \mathrm{lca}(a,b)$, and $\mathrm{lca}(a,b,c)v$ is a 1-edge. The subtree $T_{|\{abc\}}$ of (T,λ) with leaves a,b,c thus corresponds to one of T_3, T_5, T_7, T_{10}, T_{11}, or T_{12} shown in Fig. 46. Recall that T_3, T_5, T_7, and T_{10} explain the induced subgraphs A_5, A_6, A_7, and A_8, respectively. If $T_{|\{abc\}}$ is one of T_3, T_5, T_7, or T_{10}, then we would have a triple with leaves a,b,c in $\mathcal{R}(\mathcal{X})$. Since this is not the case by assumption, $T_{|\{abc\}}$ must be either T_{11} or T_{12}. Thus the subgraph of $\mathcal{X}$ induced by a,b,c is isomorphic to either A_1 or A_4.

Moreover, by Lemma 8.10, there must be a leaf $d \in \mathsf{child}(v)$ such that vd is a 0-edge. Hence, the subtrees $T_{|\{acd\}}$ and $T_{|\{bcd\}}$ with leaves a,c,d and b,c,d,

respectively, correspond to one of the trees T_3, T_5, T_7, or T_{10}. Thus the subgraph of $\mathcal{X}$ induced by a, c, d or b, c, d must be isomorphic to a valid triangle A_5, A_6, A_7, or A_8. By construction, $ad|c \in \mathcal{R}(\mathcal{X})$ and $bd|c \in \mathcal{R}(\mathcal{X})$. Hence, any tree that explains $\mathcal{X}$ must display $ad|c$ and $bd|c$. As shown in [45], a tree displaying $ad|c$ and $bd|c$ also displays $ab|c$. This implies, however, that $ab|c \in \mathrm{cl}(\mathcal{R}(\mathcal{X}))$, a contradiction to our assumption.

Therefore, $\mathrm{cl}(\mathcal{R}(\mathcal{X})) = r(T)$ and we can finally apply Lemma 3.1 to conclude that $\mathcal{R}(\mathcal{X})$ identifies (T, λ). □

We are now in the position to prove Thm. 8.1, the main result of this section:

Proof of Theorem 8.1. The first statement is an immediate consequence of Lemma 8.8. Lemma 8.11 implies that $\mathcal{R}(\mathcal{X})$ identifies $(\widehat{T}, \widehat{\lambda})$. Hence, any tree that displays $\mathcal{R}(\mathcal{X})$ is a refinement of $(\widehat{T}, \widehat{\lambda})$ and thus, must have more vertices. Lemma 8.11 also implies that (T, λ) displays $(\widehat{T}, \widehat{\lambda})$. Moreover, Lemma 8.3 ensures that any tree explaining $\mathcal{X}$ displays $\mathcal{R}(\mathcal{X})$. Combining these two observations, we conclude that $\widehat{T}$ has the minimum number of vertices among all trees that explain $\mathcal{X}$.

By Lemma 8.10, all inner and outer edges of $(\widehat{T}, \widehat{\lambda})$ are relevant, and thus, their labels cannot be changed without changing $\mathcal{X}$. Moreover, Lemma 8.9 implies that there is no further sequence of extended edge contractions that could be applied to $(\widehat{T}, \widehat{\lambda})$ in order to obtain another tree that explains $\mathcal{X}$. Hence, $(\widehat{T}, \widehat{\lambda})$ is unique. □

8.3 Characterization of valid xenology relations

This section is dedicated to the proof of the main result of this chapter: a binary relation $\mathcal{X}$ is explained by a tree if and only if it contains only valid triangles.

Theorem 8.2. *An irreflexive relation $\mathcal{X}$ on L is valid if and only if it is a Fitch relation.*

The key idea of the proof, which proceeds by induction on the number of leaves, is to consider the superposition of trees explaining two induced subrelations, each of which is obtained by removing a single vertex from $\mathcal{X}$. We first establish several technical results for these trees. To this end we introduce some notation that will be used in this section only.

Definition 8.8. *Let (T, λ) be an edge-labeled phylogenetic tree and $e = uv$ be an outer edge of T. We write $(T - v, \lambda_{|L-v})$ for the tree obtained from (T, λ) by removing the outer edge e and vertex v from T and keep the edge labels of all remaining edges.*

For an outer edge $e = uv$ we therefore have $(T - v, \lambda_{|L-v}) = (T^e, \lambda^e)$ if and only if either $u = \rho_T$ and $\deg_{T-v}(u) > 1$, or $u \neq \rho_T$ and $\deg_{T-v}(u) > 2$.

Definition 8.9. *Let $\mathcal{X} \subset L \times L$ be an irreflexive relation and consider $l_1, \ldots, l_k \in L$. The set $\mathcal{X}_{\neg l_1,\ldots,l_k}$ denotes the subrelation of $\mathcal{X}$ that is induced by $L \setminus \{l_1, \ldots, l_k\}$.*

We emphasize that the results established in the previous sections are in general not valid for non-phylogenetic trees. Nevertheless, it is useful in the following to extend some concepts to more general trees. In particular, we say that an edge-labeled rooted (but possibly non-phylogenetic) tree (T, λ) with leaf set L *explains* a given irreflexive relation $\mathcal{X} \subset L \times L$ if for any pair $(x, y) \in \mathcal{X}$ there is a 1-edge on the path from $\mathrm{lca}(x, y)$ to y.
Using the same arguments as in the proof of Lemma 8.1 we observe that $(T - v, \lambda_{|L-v})$ explains $\mathcal{X}_{\neg v}$.

Lemma 8.12. *Let (T, λ) be a least resolved phylogenetic tree on L w.r.t. $\mathcal{X} = \mathcal{X}_{(T,\lambda)}$, and $v \in L$. Let (T', λ') be a least resolved phylogenetic tree w.r.t. $\mathcal{X}_{\neg v}$. Then, (T', λ') is displayed by $(T - v, \lambda_{|L-v})$. In particular, $(T', \lambda') = (T - v, \lambda_{|L-v})$ if and only if (i) $\mathsf{par}(v) = \rho_T$ and $\deg_T(\rho_T) > 2$ or (ii) $\deg_T(\mathsf{par}(v)) > 3$ and $\lambda_{|L-v}(\mathsf{par}(v)u) = 0$ for some child $u \in \mathsf{child}(\mathsf{par}(v))$, $u \neq v$.*

Proof. If $(T - v, \lambda_{|L-v})$ is phylogenetic, then we may apply Thm. 8.1 to verify that (T', λ') is indeed displayed by $(T - v, \lambda_{|L-v})$. Now assume that $(T - v, \lambda_{|L-v})$ is not phylogenetic. In this case, either (a) $\mathsf{par}(v) \neq \rho_T$ is an inner vertex of degree 2, or (b) the root ρ of $T - v$ has degree 1, and hence $\rho_T = \mathsf{par}(v)$.

Case (a): If $x = \mathsf{par}(v) \neq \rho_T$ is an inner vertex of degree 2, let T^* be the tree obtained from $(T - v, \lambda_{|L-v})$ by a simple contraction of the edge $\mathsf{par}(x)x$ and setting $\lambda_{|L-v}(x\mathsf{child}(x)) = 1$. The labels of all other edges are kept. By construction, we obtain a phylogenetic tree (T^*, λ^*) that still explains $\mathcal{X}_{\neg v}$ and, by Thm. 8.1, satisfies $(T', \lambda') \leq (T^*, \lambda^*) \leq (T - v, \lambda_{|L-v})$. Therefore, (T', λ') is displayed by $(T - v, \lambda_{|L-v})$.

Case (b): If the root ρ of $T - v$ has degree 1, let T^* be the tree obtained by deleting ρ and the edge ρw, where w denotes the unique child of ρ in $T - v$, and declaring w as the root of T^*. For all other edges set $\lambda^*(e) = \lambda_{|L-v}(e)$. Again, we obtain a phylogenetic tree (T^*, λ^*) that still explains $\mathcal{X}_{\neg v}$. Repeating the arguments of *Case (a)*, we can conclude that (T', λ') is displayed by $(T - v, \lambda_{|L-v})$.

Now assume that $(T', \lambda') = (T - v, \lambda_{|L-v})$. There are two cases: either $\mathsf{par}(v)$ is the root ρ_T or not. If $\mathsf{par}(v) = \rho_T$, then $\deg_T(\rho_T) \leq 2$ would imply that $\deg_{T'}(\rho_T) \leq 1$, in which case (T', λ') would not be a phylogenetic tree; a contradiction since (T', λ') is phylogenetic. Hence, if $\mathsf{par}(v) = \rho_T$, then $\deg_T(\rho_T) > 2$. Now assume that $\mathsf{par}(v) \neq \rho_T$. Thus there is an inner edge $x\mathsf{par}(v)$ where $x = \mathsf{par}(\mathsf{par}(v))$. Lemma 8.10(3) implies that this edge $x\mathsf{par}(v)$ must be incident to an outer 0-edge in (T', λ') and hence, $\lambda_{|L-v}(\mathsf{par}(v)u) = 0$ for some leaf $u \in L \setminus \{v\}$. Moreover, as (T', λ') is phylogenetic, $\deg_{T-v}(\mathsf{par}(v)) > 2$ and hence, $\deg_T(\mathsf{par}(v)) > 3$.

Conversely, assume first that $\mathsf{par}(v) = \rho_T$ and $\deg_T(\rho_T) > 2$. In this case, $(T - v, \lambda_{|L-v})$ is still a phylogenetic tree. By construction, $E^0(T - v) = E^0(T)$ and $\lambda_{|L-v}(e) = \lambda(e)$ for all $e \in E^0(T - v)$ Thus any inner edge of $T - v$ is a 1-edge. Lemma 8.10(3) implies that for each inner edge $e = xy$ in T there is an outer 0-edge yz in (T, λ). This property still holds in $(T - v, \lambda_{|L-v})$ because the deleted edge $\mathsf{par}(v)v$ is incident to the root of (T, λ). Thus all edges of $(T - v, \lambda_{|L-v})$ are relevant. Lemma 8.10 implies that $(T - v, \lambda_{|L-v})$ is least resolved.

Now assume that $\mathsf{par}(v) \neq \rho_T$ and $\deg_T(\mathsf{par}(v)) > 3$. Thus $(T - v, \lambda_{|L-v})$ is still a phylogenetic tree. Moreover, let $\lambda_{|L-v}(\mathsf{par}(v)u) = 0$ for some child $u \in \mathsf{child}(\mathsf{par}(v))$, $u \neq v$. Now, we can apply similar arguments as above to conclude that all edges in $(T - v, \lambda_{|L-v})$ are relevant, and thus $(T - v, \lambda_{|L-v})$ is least resolved.

In summary, if $\mathsf{par}(v) = \rho_T$ and $\deg_T(\rho_T) > 2$ or $\lambda_{|L-v}(\mathsf{par}(v)u) = 0$ for some child $u \in \mathsf{child}(\mathsf{par}(v))$, $u \neq v$, and $\deg_T(\mathsf{par}(v)) > 3$, then $(T - v, \lambda_{|L-v})$ is least resolved w.r.t. $\mathcal{X}_{\neg v}$. By Thm. 8.1, $(T - v, \lambda_{|L-v}) = (T', \lambda')$. □

An immediate consequence of Lemma 8.12 is the following result that is crucial for proving the main result.

Lemma 8.13. *Let (T, λ) and $(T - v, \lambda_{|L-v})$ be defined as in Lemma 8.12, and (T', λ') be the least resolved phylogenetic tree that explains $\mathcal{X}_{\neg v}$. Then, either*

1. *$(T - v, \lambda_{|L-v}) = (T', \lambda')$, or*
2. *(T', λ') is obtained from $(T - v, \lambda_{|L-v})$ by a simple contraction of either*
   - *(i) the inner edge $\rho_T u \in E(T - v)$, in case that $\mathsf{par}(v) = \rho_T$ and $\deg_T(\rho_T) = 2$, or*
   - *(ii) the inner edge $\mathsf{par}(x)x \in E(T - v)$, where $x = \mathsf{par}(v) \neq \rho_T$, and setting $\lambda'(x\mathsf{child}(x)) = 1$, otherwise.*

   In either case $\lambda'(e) = \lambda_{|L-v}(e)$ for all non-contracted edges e.

In particular, $(T - v, \lambda_{|L-v})$ displays the least resolved phylogenetic tree (T', λ') that explains $\mathcal{X}_{\neg v}$ and therefore, $r(T') \subseteq r(T - v)$.

Proof. By Lemma 8.12, $(T - v, \lambda_{|L-v})$ is least resolved if and only if $\mathsf{par}(v) = \rho_T$ and $\deg_T(\rho_T) > 2$, or there exists a leaf $u \in \mathsf{par}(v)$, $u \neq v$, such that $\lambda_{|L-v}(\mathsf{par}(v)u) = 0$ and $\deg_T(\mathsf{par}(v)) > 3$. If $(T - v, \lambda_{|L-v})$ is not least resolved and $\mathsf{par}(v) = \rho_T$, we have $\deg_{T-v}(\rho_T) = 1$. Due to Lemma 8.10(6), the tree (T', λ') obtained by a simple contraction of the single edge $\rho_T u$ and adopting u as the new root is least resolved w.r.t. $\mathcal{X}_{\neg v}$.

If $(T - v, \lambda_{|L-v})$ is not least resolved and $\mathsf{par}(v) \neq \rho_T$, then either (a) there is no leaf $u \in \mathsf{child}(\mathsf{par}(v))$, $u \neq v$, with $\lambda_{|L-v}(\mathsf{par}(v)u) = 0$, or (b) $\deg_{T-v}(\mathsf{par}(v)) = 2$. Indeed, $\deg_{T-v}(\mathsf{par}(v)) > 2$ and $u \in \mathsf{child}(\mathsf{par}(v))$ with $\lambda_{|L-v}(\mathsf{par}(v)u) = 0$ implies that $(T - v, \lambda_{|L-v})$ is least resolved. On the other hand, $\deg_{T-v}(\mathsf{par}(v)) \geq 2$ because T is phylogenetic.

Case (a). Assume that $\mathsf{par}(v)u$ is a 1-edge for all children $u \neq v$ of $\mathsf{par}(v)$. Then the inner edge $\mathsf{par}(x)x \in E(T - v)$ is irrelevant in $(T - v, \lambda_{|L-v})$; thus it can be *ex*-contracted. Since (T, λ) is least resolved, Lemma 8.10(3) ensures that every inner vertex in $(T - v, \lambda_{|L-v})$ other than $\mathsf{par}(v)$ is adjacent to an outer 0-edge. Hence, extended contraction of $x\mathsf{par}(v)$ in $(T - v, \lambda - v)$ yields the least resolved tree w.r.t. $\mathcal{X}_{\neg v}$.

Case (b). If $\deg_{T-v}(\mathsf{par}(v)) = 2$ and $\lambda_{|L-v}(\mathsf{par}(v)u) = 1$, the edge $\mathsf{par}(x)x$ can be *ex*-contracted without changing the relation and similar arguments as in Case (a) show that $(T - v, \lambda_{|L-v})$ is least resolved w.r.t. $\mathcal{X}_{\neg v}$. If $\lambda_{|L-v}(\mathsf{par}(v)u) = 0$, then the construction as in Property 2.(ii) does not change

$\mathcal{X}$ since $\lambda(\mathsf{par}(x)x) = 1$. Again, similar arguments as in Case (a) ensure that $(T - v, \lambda_{|L-v})$ is least resolved w.r.t. $\mathcal{X}_{\neg v}$.

Obviously, it either holds $(T', \lambda') = (T - v, \lambda_{|L-v})$ or (T', λ') can be obtained from $(T - v, \lambda_{|L-v})$ by a single simple edge contraction. Thus (T', λ') is displayed by $(T - v, \lambda_{|L-v})$ and $r(T') \subseteq r(T - v)$. □

Let $(T = (V, E), \lambda)$ be an edge-labeled phylogenetic tree. Moreover, let $xy \in E$ and let (T^e, λ^e) be the phylogenetic tree obtained from (T, λ) by extended contraction of e in (T, λ). Given (T^e, λ^e) it is possible to recover the tree (T, λ) reverting the extended contraction of e. If e was an internal edge, this amounts to subdividing a vertex z, yielding $e = uv$, and a bi-partitioning of the set of children of z into the children of u and v. If e was an external edge incident to a degree 2 node, an edge f in (T^e, λ^e) is subdivided and e is attached to the new inner vertex. In addition, the labeling is adjusted. We refer to these constructions as *reinsertion of e into (T^e, λ^e)*.

Lemma 8.14. *Given a Fitch relation $\mathcal{X} \subset L \times L$, $|L| > 3$, such that $\mathcal{X}_{\neg u}$, $\mathcal{X}_{\neg v}$, and $\mathcal{X}_{\neg uv}$ are valid for some $u, v \in L$. Let $(T_{\neg u}, \lambda_{\neg u})$, $(T_{\neg v}, \lambda_{\neg v})$, and $(T_{\neg uv}, \lambda_{\neg uv})$ be the least resolved trees that explain $\mathcal{X}_{\neg u}$, $\mathcal{X}_{\neg v}$, and $\mathcal{X}_{\neg uv}$, respectively.*

Then there is a tree (T, λ) that correctly explains all members in $\mathcal{X} \setminus \mathcal{X}[u, v]$, i.e., $\mathcal{X}_{(T,\lambda)}[a, b] = \mathcal{X}[a, b]$ for all $a, b \in L$ with $\{a, b\} \neq \{u, v\}$. Moreover (T, λ) displays $(T_{\neg u}, \lambda_{\neg u})$, $(T_{\neg v}, \lambda_{\neg v})$, and $(T_{\neg uv}, \lambda_{\neg uv})$.

Proof. Consider the least resolved tree $(T_{\neg uv}, \lambda_{\neg uv})$ that explains $\mathcal{X}_{\neg uv}$. By Lemma 8.13, this tree can be obtained from the least resolved trees $(T_{\neg u}, \lambda_{\neg u})$ and $(T_{\neg v}, \lambda_{\neg v})$ by removing the vertices v and u, respectively, and possibly (simple) contraction of edges. More precisely, it either holds $(T_{\neg uv}, \lambda_{\neg uv}) = (T_{\neg u} - v, \lambda_{\neg u|L'})$, where $L' = L \setminus \{u, v\}$, or $(T_{\neg uv}, \lambda_{\neg uv})$ is obtained from $(T_{\neg u} - v, \lambda_{\neg u|L'})$ by (i) contracting $\mathsf{par}(v)w$ if $\mathsf{par}(v)$ is the root of $T_{\neg u}$ with degree 2, or otherwise (ii) contracting exactly the edge xy where $y = \mathsf{par}(v)$ and a possible relabeling of the incident edges below y (cf. Lemma 8.13). In the following we denote by w_{xy} the vertex in $T_{\neg uv}$ that is obtained by contraction of this edge xy. Moreover, we will throughout this proof refer to those two cases as contractions of Type (i) and (ii). In the same way, $(T_{\neg uv}, \lambda_{\neg uv})$ is obtained from $(T_{\neg v} - u, \lambda_{\neg v|L'})$ and if the edge $x'y'$ was contracted, then $w'_{x'y'}$ denotes the resulting vertex in $T_{\neg uv}$.
Therefore, the following cases must be considered:

1. $(T_{\neg uv}, \lambda_{\neg uv}) = (T_{\neg u} - v, \lambda_{\neg u|L'}) = (T_{\neg v} - u, \lambda_{\neg v|L'})$,
2. Either
   (a) $(T_{\neg uv}, \lambda_{\neg uv}) = (T_{\neg u} - v, \lambda_{\neg u|L'}) \lneq (T_{\neg v} - u, \lambda_{\neg v|L'})$, or
   (b) $(T_{\neg uv}, \lambda_{\neg uv}) = (T_{\neg v} - u, \lambda_{\neg v|L'}) \lneq (T_{\neg u} - v, \lambda_{\neg u|L'})$,
3. $(T_{\neg uv}, \lambda_{\neg uv}) \lneq (T_{\neg u} - v, \lambda_{\neg u|L'})$ and $(T_{\neg uv}, \lambda_{\neg uv}) \lneq (T_{\neg v} - u, \lambda_{\neg v|L'})$, where in at least one of the two cases $(T_{\neg uv}, \lambda_{\neg uv})$ can be obtained by a contraction of Type (i),

4. $(T_{\neg uv}, \lambda_{\neg uv}) \lesssim (T_{\neg u} - v, \lambda_{\neg u|L'})$ and $(T_{\neg uv}, \lambda_{\neg uv}) \lesssim (T_{\neg v} - u, \lambda_{\neg v|L'})$, where a contraction of Type (ii) has to be applied in both cases, and either
   (a) $w_{xy} \neq w'_{x'y'}$ or (b) $w_{xy} = w'_{x'y'}$.

In *Case 1*, one can simply add the edges $\mathsf{par}(v)v$ and $\mathsf{par}(u)u$ together with the original edge labels $\lambda_{\neg u}(\mathsf{par}(v)v)$ and $\lambda_{\neg v}(\mathsf{par}(u)u)$ to obtain a tree (T, λ) that contains both $(T_{\neg u}, \lambda_{\neg u})$ and $(T_{\neg v}, \lambda_{\neg v})$ as subtrees and thus, $\mathcal{X}_{(T,\lambda)}[a, b] = \mathcal{X}[a, b]$ for all a, b with $\{a, b\} \neq \{u, v\}$.

Case 2(a). Suppose first that $(T_{\neg uv}, \lambda_{\neg uv})$ is obtained from $(T_{\neg v} - u, \lambda_{\neg v|L'})$ by a contraction of Type (i). Reverting this contraction by inserting a new root ρ_T to $T_{\neg uv}$, i.e., inserting the edge $\rho_T\rho_{T_{\neg uv}}$ with label 1, and inserting the edges $\rho_T u$ with its original label $\lambda_{\neg v}(\mathsf{par}(u)u)$, yields the tree $(T_{\neg v}, \lambda_{\neg v})$. Since, by construction, we have $(T_{\neg v}(\rho_{T_{\neg uv}}), \lambda_{\neg v|L'}) = (T_{\neg uv}, \lambda_{\neg uv})$, $\mathcal{X}_{\neg uv}$ and $\mathcal{X}_{\neg v}$ are clearly explained by $(T_{\neg v}, \lambda_{\neg v})$. Moreover, as $(T_{\neg uv}, \lambda_{\neg uv}) = (T_{\neg u} - v, \lambda_{\neg u|L'})$, one can simply add the edge $\mathsf{par}(v)v$ together with the original edge label $\lambda_{\neg u}(\mathsf{par}(v)v)$ and obtains a tree (T, λ) with $(T(\rho_{T_{\neg uv}}), \lambda_{|L-u}) = (T_{\neg u}, \lambda_{\neg u})$. Hence, (T, σ) explains $\mathcal{X}_{\neg u}$ and one easily checks that it also explains $\mathcal{X}_{\neg uv}$ and $\mathcal{X}_{\neg v}$.

Now assume that $(T_{\neg uv}, \lambda_{\neg uv})$ is obtained from $(T_{\neg v} - u, \lambda_{\neg v|L'})$ by a contraction of Type (ii). Again, one can simply add the edge $\mathsf{par}(v)v$ together with the original edge label $\lambda_{\neg u}(\mathsf{par}(v)v)$ to $(T_{\neg uv}, \lambda_{\neg uv})$ in order to obtain $(T_{\neg u}, \lambda_{\neg u})$. Since $w'_{x'y'}$ denotes the vertex that results from contracting the edge $x'y'$ in $(T_{\neg v} - u, \lambda_{\neg v|L'})$, this vertex is also contained in $(T_{\neg u}, \lambda_{\neg u})$. Now, we reinsert the edge $x'y'$ in $(T_{\neg u}, \lambda_{\neg u})$ such that we obtain a tree (T, λ) that contains $(T_{\neg v}, \lambda_{\neg v})$ as a subtree. Hence, $\mathcal{X}_{\neg uv}$ and $\mathcal{X}_{\neg v}$ are correctly explained by (T, λ). It remains to show that also all $\mathcal{X}[v, z]$ and $\mathcal{X}[z, v]$ with $z \neq u$ are still correctly explained. Assume for contradiction that this is not the case and that $\mathcal{X}[v, z] \neq \mathcal{X}_{(T,\lambda)}[v, z]$ for some $z \neq u$. This is only possible if in the tree (T, λ) the 1-edge $x'y'$ contained in the path from $\mathrm{lca}_T(v, z)$ to z. Hence, $\mathcal{X}_{(T,\lambda)}[v, z] = (v, z)$, which implies that the path from $\mathrm{lca}_{T_{\neg u}}(v, z)$ to z contains only 0-edges. Moreover, $(T_{\neg u}, \lambda_{\neg u})$ is least resolved w.r.t. $\mathcal{X}_{\neg u}$. Hence, all inner edges are 1-edges. Therefore $\mathrm{lca}_{T_{\neg u}}(v, z) = w'_{x'y'}$ and $w'_{x'y'}z \in E(T_{\neg u})$ must be an outer 0-edge. Note that this implies that z is a child of y' in $T_{\neg v}$. By construction according to Lemma 8.13(ii), we have contracted the edge $x'y'$ in $(T_{\neg v} - u, \lambda_{|L-u})$ and relabeled all outer edges in $T_{\neg v}$ incident to y' as 1-edges. But this implies that $w'_{x'y'}z$ is a 1-edge in $(T_{\neg u}, \lambda_{\neg u})$; a contradiction. The assumption $\mathcal{X}[z, v] \neq \mathcal{X}_{(T,\lambda)}[z, v]$ for some $z \neq u$ yields a contradiction using analogous arguments.

Case 2(b) is settled by interchanging the roles of u and v in Case 2(a).

Case 3. Assume first that, w.l.o.g., $(T_{\neg uv}, \lambda_{\neg uv})$ is obtained by a contraction of Type (i) from $(T_{\neg v} - u, \lambda_{\neg v|L'})$ and of Type (ii) from $(T_{\neg u} - v, \lambda_{\neg u|L'})$. First, we revert the contraction of Type (ii) by reinserting the edge xy and yv with their original labels. This yields the tree $(T_{\neg u}, \lambda_{\neg u})$. Then we insert a new root ρ_T to $T_{\neg u}$, i.e., inserting the edge $\rho_T\rho_{T_{\neg u}}$ with label 1, as well as the edge $\rho_T u$ with its original edge label. The resulting tree is denoted by (T, λ). Clearly, (T, λ) displays $(T_{\neg uv}, \lambda_{\neg uv})$ and $(T_{\neg u}, \lambda_{\neg u})$, and explains $\mathcal{X}_{\neg uv}$ and $\mathcal{X}_{\neg u}$. It

remains to show that $\mathcal{X}[u,z]$ is correctly explained by (T,λ) for all $z \in L'$. By construction, u is incident to the root in $(T_{\neg v},\lambda_{\neg v})$ and the degree of this root is 2. Since $(T_{\neg v},\lambda_{\neg v})$ is least resolved and $|L|>3$, this only remaining child of this root must be an inner 1-edge. Hence, $(u,z)\in X$ for any $z\in L'$. Moreover, for each $z\in L'$, we have $(z,u)\in X$ if and only if $\lambda_{\neg v}(\mathsf{par}(u)u)=1$. One now easily checks that, by construction, (T,σ) explains $\mathcal{X}[u,z]$ for any $z\in L'$.

If $(T_{\neg uv},\lambda_{\neg uv})$ is obtained by a contraction of the form (i) from both $(T_{\neg u}-v,\lambda_{\neg u|L'})$ and $(T_{\neg v}-u,\lambda_{\neg v|L'})$, we construct a tree (T,λ) by adding a new root ρ_T to $T_{\neg uv}$, i.e., inserting the edge $\rho_T\rho_{T_{\neg uv}}$ with label 1, and inserting the edges $\rho_T u$ and $\rho_T v$ with their original edge labels. Clearly, $(T(\rho_{T_{\neg uv}}),\lambda_{|L'})=(T_{\neg uv},\lambda_{\neg uv})$ and thus, (T,σ) explains $\mathcal{X}_{\neg uv}$. Applying analogous arguments as used in the first part of Case 3, one easily checks that it also explains $\mathcal{X}[u,z]$ and $\mathcal{X}[v,z]$ for any $z\in L'$.

Case 4. In order to obtain (T,λ) from $(T_{\neg uv},\lambda_{\neg uv})$, we need to undo the contractions that lead to w_{xy} and $w'_{x'y'}$ and in addition, reinsert the edges $y'u$ and yv with original edge labeling such that (T,λ) contains both $(T_{\neg u},\lambda_{\neg u})$ and $(T_{\neg v},\lambda_{\neg v})$ as subtrees and thus, $\mathcal{X}_{(T,\lambda)}[a,b]=\mathcal{X}[a,b]$ for all a,b with $\{a,b\}\neq\{u,v\}$. The subdivision of w_{xy} partitions the set of children $\mathsf{child}(w_{xy})$ of the vertex w_{xy} into two disjoint sets $\mathfrak{C}_x$ and $\mathfrak{C}_y$ in such a way that $\mathfrak{C}_x$ contains all children of x that are distinct from y and $\mathfrak{C}_y$ contains all children of y in $(T_{\neg u}-v,\lambda_{\neg u|L'})$. Analogously, the sets $\mathfrak{C}_{x'}$ and $\mathfrak{C}_{y'}$ are obtained by partitioning $\mathsf{child}(w'_{x'y'})$ in $(T_{\neg v}-u,\lambda_{\neg v|L'})$. The sets $\mathfrak{C}_x$, $\mathfrak{C}_{x'}$, $\mathfrak{C}_y$, and $\mathfrak{C}_{y'}$ are all non-empty because $(T_{\neg u},\lambda_{\neg u})$ and $(T_{\neg v},\lambda_{\neg v})$ are phylogenetic.

Case 4(a). $w_{xy}\neq w'_{x'y'}$. By definition of $(T_{\neg uv},\lambda_{\neg uv})$, it is possible to subdivide w_{xy} and add $\mathsf{par}(v)v$ with the edge labeling $\lambda_{\neg u}(\mathsf{par}(v)v)$ such that we obtain $(T_{\neg u},\lambda_{\neg u})$. Subdivision of $w'_{x'y'}$ in $(T_{\neg u},\lambda_{\neg u})$ results in a tree (T,λ) that contains $(T_{\neg v},\lambda_{\neg v})$ as a subtree. Hence, (T,λ) correctly explains $\mathcal{X}_{\neg uv}$ and $\mathcal{X}_{\neg v}$. Arguments analogous to Case 2 now show that $\mathcal{X}[z,v]$ and $\mathcal{X}[v,z]$ are correctly explained for any $z\neq u$, thus (T,λ) correctly explains $\mathcal{X}_{\neg u}$.

Case 4(b). $w_{xy}=w'_{x'y'}$. Since $w_{xy}=w'_{x'y'}$, (T,λ) is obtained from $(T_{\neg uv},\lambda_{\neg uv})$ by reinsertion of a single edge. To ensure that (T,λ) displays both $(T_{\neg u},\lambda_{\neg u})$ and $(T_{\neg v},\lambda_{\neg v})$, we need to show that $\mathfrak{C}_x=\mathfrak{C}_{x'}$ and $\mathfrak{C}_y=\mathfrak{C}_{y'}$.

First, we show that all 0-edges incident to w_{xy} in $(T_{\neg uv},\lambda_{\neg uv})$ are incident to x and x' in $(T_{\neg u},\lambda_{\neg u})$ and $(T_{\neg v},\lambda_{\neg v})$, respectively. Let M denote the set of all leaves $z\in\mathsf{child}(w_{xy})$ for which $\lambda'(w_{xy}z)=0$ in $T_{\neg uv}$. Since $(T_{\neg uv},\lambda_{\neg uv})$ is least resolved, $M\neq\emptyset$. For any $w\in\mathsf{child}(w_{xy})$, and $z\in M$ there is no 1-edge on the path from $\mathrm{lca}(w,z)$ to z in $(T_{\neg uv},\lambda_{\neg uv})$. We proceed by showing that $M\subseteq\mathfrak{C}_x\cap\mathfrak{C}_{x'}$. Assume for contradiction that $z\in\mathfrak{C}_x$ but $z\notin\mathfrak{C}_{x'}$. Thus $z\in\mathfrak{C}_{y'}$. Furthermore, for any $w'\in\mathfrak{C}_{x'}$, the 1-edge $e'=x'y'$ is contained in the path from $\mathrm{lca}(w',z)$ to z in the tree $(T_{\neg v}-u,\lambda_{v|L'})$. Since $(T_{\neg v}-u,\lambda_{v|L'})$ is phylogenetic, $\mathfrak{C}_{x'}$ is non-empty, i.e., such a w' exists. In contrast, for any $w\in\mathfrak{C}_x\cup\mathfrak{C}_y$, $w\neq z$, there is no 1-edge on the path from $\mathrm{lca}(w,z)$ to z in $(T_{\neg u}-v,\lambda_{u|L'})$. Since $\mathfrak{C}_{x'}\subseteq\mathfrak{C}_x\cup\mathfrak{C}_y$, the two trees $(T_{\neg u}-v,\lambda_{u|L'})$ and $(T_{\neg v}-u,\lambda_{v|L'})$ cannot explain the same relation $\mathcal{X}_{\neg uv}$; this is the desired contradiction.

Hence, it remains to show that for every 1-edge $w_{xy}w''$ in $(T_{\neg uv},\lambda_{\neg uv})$ either $w''\in\mathfrak{C}_x\cap\mathfrak{C}_{x'}$ or $w''\in\mathfrak{C}_y\cap\mathfrak{C}_{y'}$ is true. Assume for contradiction that $w''\in\mathfrak{C}_x$ but $w''\notin\mathfrak{C}_{x'}$, i.e., $w''\notin\mathfrak{C}_y$ and $w''\in\mathfrak{C}_{y'}$. This implies $[w'',v]\in\mathcal{X}_{\neg u}$ and

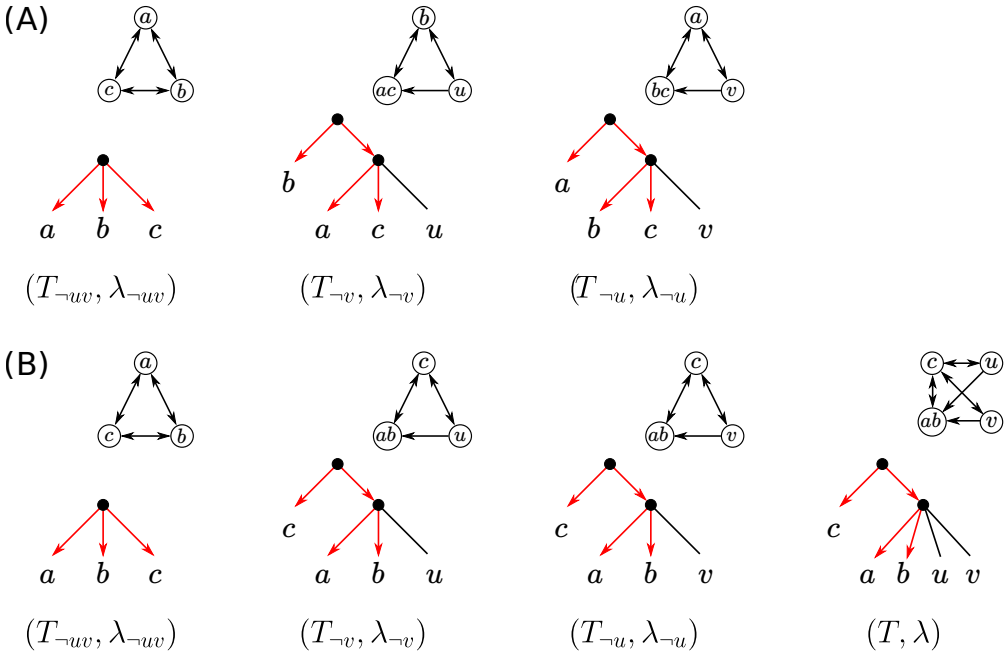


Fig. 48. (A) The two least resolved trees $(T_{\neg u}, \lambda_{\neg u})$ and $(T_{\neg v}, \lambda_{\neg v})$ that explain $\mathcal{X}_{\neg u}$ and $\mathcal{X}_{\neg v}$ respectively both explain the least resolved tree $(T_{\neg uv}, \lambda_{\neg uv})$ that explains $\mathcal{X}_{\neg uv}$. However, there exists no tree (T, λ) that explains $\mathcal{X}$, thus there is no valid Fitch relation $\mathcal{X}$ that contains both $\mathcal{X}_{\neg u}$ and $\mathcal{X}_{\neg v}$. This is due to the fact that the triples $ac|b$ and $bc|a$ in $(T_{\neg u}, \lambda_{\neg u})$ and $(T_{\neg v}, \lambda_{\neg v})$ contradict each other. (B) We have $\mathfrak{C}_x = \{c\} = \mathfrak{C}_{x'}$ and $\mathfrak{C}_y = \{a, b\} = \mathfrak{C}_{y'}$ in the least resolved trees $(T_{\neg u}, \lambda_{\neg u})$ and $(T_{\neg v}, \lambda_{\neg v})$. In this case, there exists a tree (T, λ) that displays $(T_{\neg uv}, \lambda_{\neg uv})$, $(T_{\neg u}, \lambda_{\neg u})$ and $(T_{\neg v}, \lambda_{\neg v})$, and explains $\mathcal{X}$. The Fitch relation corresponding to each tree is shown in the upper right corner. Two nodes x and y are represented as one node xy if they have the same relationship with every other node.

$(u, w'') \in \mathcal{X}_{\neg v}$. Since $\{u, v, w''\}$ must form a valid triangle, either $(u, v) \in \mathcal{X}$ or $[u, v] \in \mathcal{X}$ must be true. On the other hand, since $M \subseteq \mathfrak{C}_x \cap \mathfrak{C}_{x'}$ and the trees $(T_{\neg u}, \lambda_{\neg u})$ and $(T_{\neg v}, \lambda_{\neg v})$ are least resolved, both yv and $y'u$ must be 0-edges. By construction, (T, λ) is obtained by reinserting a single edge in $(T_{\neg uv}, \lambda_{\neg uv})$ in such a way that $\mathsf{par}(u) = \mathsf{par}(v)$. Thus we must have $u|v$; a contradiction. We therefore conclude $\mathfrak{C}_y = \mathfrak{C}_{y'}$ and $\mathfrak{C}_x = \mathfrak{C}_{x'}$, which completes the proof. □

We remark that the existence of the tree (T, λ) asserted in Lemma 8.14 does not follow from the fact that both $(T_{\neg u}, \lambda_{\neg u})$ and $(T_{\neg v}, \lambda_{\neg v})$ explain $(T_{\neg uv}, \lambda_{\neg uv})$. A counterexample is given in Fig. 48. The condition that the trees together explain a Fitch relation cannot be relaxed in the proof.

We are now in the position to prove the main result of this section.

Proof of Theorem 8.2. Assume that $\mathcal{X}$ is valid. Hence, there is a tree (T, λ) that explains $\mathcal{X}$. Let $x, y, z \in L$ be distinct vertices. Clearly, any subtree $T' \subseteq T$ with leaf set $\{x, y, z\}$ must correspond to one of the trees $T_1, \ldots, T_{16}$ in Fig. 46. Since these subtrees can only encode the valid triangles $A_1, \ldots A_8$, the subgraph induced by x, y, z in $\mathcal{X}$ must be isomorphic to one of $A_1, \ldots A_8$. As

this statement is true for any three distinct vertices in $\mathcal{X}$, all triangles in $\mathcal{X}$ are valid. Hence, $\mathcal{X}$ is a Fitch relation.

Now assume that $\mathcal{X}$ is a Fitch relation. The trivial relation on L, corresponding to the empty graph, is explained by any tree with leaf set L that has only 0-edges. For the non-trivial case we proceed by induction w.r.t. the number of vertices $|L|$. The base case consists of the valid triangles, for which the statement is trivially true. Assume now that all Fitch relations with $|L| \leq n$ are valid.

Let $\mathcal{X}$ be a Fitch relation on $|L| = n + 1$ vertices and let $u, v \in L$ be two distinct, arbitrarily chosen vertices. Clearly, $\mathcal{X}_{\neg u}$, $\mathcal{X}_{\neg v}$, and $\mathcal{X}_{\neg uv}$ are Fitch relations and, by assumption, also valid. In particular, there are unique least resolved trees $(T_{\neg u}, \lambda_{\neg u})$, $(T_{\neg v}, \lambda_{\neg v})$, and $(T_{\neg uv}, \lambda_{\neg uv})$ that explain $\mathcal{X}_{\neg u}$, $\mathcal{X}_{\neg v}$, and $\mathcal{X}_{\neg uv}$, respectively. With the exception of the relation between u and v, $\mathcal{X}$ is therefore determined by $(T_{\neg u}, \lambda_{\neg u})$ and $(T_{\neg v}, \lambda_{\neg v})$, i.e., any pair $(x, y) \in \mathcal{X}$ for which $\{x, y\} \neq \{u, v\}$ is explained by $(T_{\neg u}, \lambda_{\neg u})$ or $(T_{\neg v}, \lambda_{\neg v})$. In particular all pairs (x, u) or (u, x) in $\mathcal{X} \setminus \mathcal{X}[u, v]$ are explained by $(T_{\neg v}, \lambda_{\neg v})$ and all pairs (x, v) or (v, x) in $\mathcal{X} \setminus \mathcal{X}[u, v]$ are explained by $(T_{\neg u}, \lambda_{\neg u})$.

Lemma 8.14 implies that there is a tree that correctly explains all pairs in $\mathcal{X} \setminus \mathcal{X}[u, v]$ and displays $(T_{\neg u}, \lambda_{\neg u})$, $(T_{\neg v}, \lambda_{\neg v})$, and $(T_{\neg uv}, \lambda_{\neg uv})$. Thus there is in particular a least resolved tree (T, λ) that fulfills these requirements.

$\mathcal{X}[u, v]$ is in some cases uniquely determined by $\mathcal{X} \setminus \mathcal{X}[u, v]$ and the requirement that $\{u, v, x\}$ forms a valid triangle. The existence of (T, λ) then implies immediately that $\mathcal{X}[u, v]$ and hence $\mathcal{X}$, is explained by (T, λ). This is not always the case, however. If more than one choice of $\mathcal{X}[u, v]$ completes $\mathcal{X} \setminus \mathcal{X}[u, v]$, we need to show that a (T, λ) exists for each of the possible choices. Denote by Δ_{uv} the set of triangles in $\mathcal{X}$ that contain u and v. Full enumeration (which we leave to the reader) shows that $\mathcal{X}[u, v]$ is not uniquely determined if and only if all triangles in Δ_{uv} are of the form A, B, C, or D listed in Fig. 49. Only certain combinations of these triangle types can occur: The co-occurence of A and B implies $(u, v) \in \mathcal{X}$, thus $\mathcal{X}[u, v]$ is uniquely determined and hence, (T, λ) is also unique. The remaining cases can be classified as follows:

1. Δ_{uv} contains at least one triangle of each of the Types A, C, and D but not B, or of each of the Types B, C, and D but not A.

2. Δ_{uv} consists of triangles of exactly one of the Types A, B and C, D, respectively, and for each type there is a triangle.

3. Δ_{uv} consists exclusively of triangles of the Types C and D and for each Type C, D there is a triangle.

4. All triangles in Δ_{uv} are of the same type.

In each of these cases, there is more than one possible choice for $\mathcal{X}[u, v]$. Lemma 8.14 ensures that there is a least resolved tree (T, λ) that explains at least one of these choices. Given (T, λ) for one particular choice, we show below that it is always possible to transform (T, λ) into another least resolved tree that explains $\mathcal{X}$ with a different choice of $\mathcal{X}[u, v]$. The resulting tree (T', λ') is unique by Thm. 8.1 and thus, the transformation can be inverted in a uniquely defined manner.

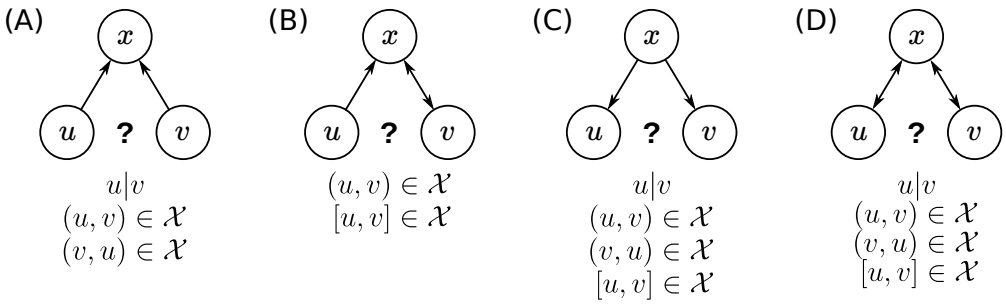


Fig. 49. All cases where the relationship $\mathcal{X}[u, v]$ cannot be uniquely inferred from $\mathcal{X} \setminus \mathcal{X}[u, v]$.

In the following we call $v \in L(T)$ a *sink* (resp. *source*) if for all $x \in L(T)$ we have $(x, v) \in \mathcal{X}_{(T,\lambda)}$ (resp. $(v, x) \in \mathcal{X}_{(T,\lambda)}$). Moreover, in order to exclude the trivial case $\Delta_{uv} = \emptyset$, we assume that $|L(T)| \geq 3$.

Case 1a. Suppose that Δ_{uv} contains at least one triangle of each of the Types A, C, and D but not of Type B. Hence, $\mathcal{X}[u, v] \in \{(u, v), (v, u), u|v\}$. Thus we show that for each of these choices of $\mathcal{X}[u, v]$ there is a least resolved tree that explains $\mathcal{X}$.

Suppose that (T, λ) explains $u|v$. Since all inner edges of (T, λ) are 1-edges, u and v must be siblings and in particular, the edges $\mathrm{lca}(u, v)v$ and $\mathrm{lca}(u, v)u$ are 0-edges. Since each triangle containing u and v is of Type A, C, or D, there is no leaf $x \in L(T) \setminus \{u, v\}$ with $x|u$ or $x|v$. Moreover, if there would be another vertex $x \in V(T) \setminus \{u, v\}$ that is adjacent to $\mathrm{lca}_T(u, v)$, then $\mathrm{lca}(u, v)x$ must be a 1-edge. Let T^* denote the subtree of T with root $\mathrm{lca}(u, v)$ without the leaves u and v. Then (T, λ) locally looks like the tree shown in the first panel in Fig. 50(1a). In order to obtain a tree that explains (u, v), we can modify (T, λ) locally to obtain a tree (T', λ') by inserting a single inner 1-edge ab in such a way that b becomes the new root of T^* and u is adjacent to a and v adjacent to b in (T', λ'). Thus u and v are not siblings anymore. Moreover, we keep all edge labelings and set $\lambda'(au) = \lambda'(bv) = 0$. By construction, $\mathcal{X}[u, v]_{(T',\lambda')} = \{(u, v)\}$. We note that $\mathrm{lca}(u, v)$ cannot be the root of T since we have a triangle of the form D, i.e., there must be an inner 1-edge ancestral to $\mathrm{lca}(u, v)$. One easily checks that (T', λ') still explains all remaining pairs in $\mathcal{X} \setminus \mathcal{X}[u, v]$. Hence, (T', λ') explains $\mathcal{X}$ whenever $\mathcal{X}[u, v] = \{(u, v)\}$. It is least resolved by construction and Lemma 8.10, and thus unique by Thm. 8.1.
Analogously, a tree (T', λ') that explains $\mathcal{X} \setminus \mathcal{X}[u, v]$ with $\mathcal{X}_{T,\lambda}[u, v] = (v, u)$ can be obtained from (T, λ) by interchanging the roles of u and v.

Finally, whenever (T', λ') explains either (u, v) or (v, u) we can obtain a tree (T, λ) that explains $u|v$ by "reversing" the contraction above. Because of the uniqueness of (T', λ') it *must* locally look as in Fig. 50(1a) middle. That is, there is exactly *one* inner 1-edge along the path from u to v and all edges incident to $\mathsf{par}(v)$ must be 1-edges. Hence, after collapsing this edge to a single vertex, we obtain the least resolved tree (T, λ) that explains $u|v$. Since $\mathcal{X}_{(T',\lambda')}[u, z] = \mathcal{X}_{(T,\lambda)}[u, z]$ and $\mathcal{X}_{(T',\lambda')}[v, z] = \mathcal{X}_{(T,\lambda)}[v, z]$ is still true for all $z \in L(T')$, (T', λ') explains $\mathcal{X}$ whenever $\mathcal{X}[u, v] = u|v$.

Case 1b. Suppose Δ_{uv} contains at least one triangle of each of the Types B, C, and D but not of Type A. Then, $\mathcal{X}[u,v] \in \{(u,v),[u,v]\}$. If (T,λ) explains $[u,v]$, it has the following properties: Since Δ_{uv} contains triangles of Type B, v but not u is a sink, and therefore $\mathsf{par}(v)v$ is a 1-edge while $\mathsf{par}(u)u$ is a 0-edge. Moreover, since $(v,u) \in \mathcal{X}_{(T,\lambda)}$ and $\mathsf{par}(u)u$ is a 0-edge, the path from $\mathrm{lca}(u,v)$ to u has to contain at least one 1-edge, u and v cannot have the same parent, thus $\mathrm{lca}(u,v) \succ_T \mathsf{par}(u)$. The presence of only triangles of Type B, C, and D immediately implies that $(u,x) \in \mathcal{X}$ if and only if $(v,x) \in \mathcal{X}$ for all $x \in L(T) \setminus \{u,v\}$. Therefore, since each inner vertex of (T,λ) (except possibly the root) must be connected to an outer 0-edge (Lemma 8.10(3b)), there cannot be any other inner vertex on the path from $\mathrm{lca}(u,v)$ to $\mathsf{par}(u)$, hence the inner edge $\mathrm{lca}(u,v)\mathsf{par}(u)$ must be present in (T,λ). On the other hand, there must be a 0-edge $\mathsf{par}(v)z$ with $z \in L(T) \setminus \{u,v\}$ (Lemma 8.10(3b)), thus $(v,z) \notin \mathcal{X}$. As $(v,z) \in \mathcal{X}$ if and only if $(u,z) \in \mathcal{X}$, this implies $(u,z) \notin \mathcal{X}$. Hence, the path from $\mathrm{lca}(u,v)$ to $\mathsf{par}(v)$ cannot contain 1-edges and therefore $\mathrm{lca}(u,v) = \mathsf{par}(v)$. Moreover, Types B, C, and D imply that there may be other 0- or 1-edges incident to $\mathsf{par}(v)$. We denote by T^{**} the subtree rooted at $\mathsf{par}(u)$ that does not contain the leaf u. The subtree of T that is rooted at $\mathsf{par}(v)$ but does neither contain the leaf v nor the leaf u nor any of the vertices of T^{**} is denoted by T^*. Thus (T,λ) must match the pattern shown Fig. 50(1b, left).

A tree (T',λ') that explains $\mathcal{X}$ with $\mathcal{X}[u,v] = (u,v)$ can now be constructed by a simple change in the position of v in (T,λ), that is, we delete the 1-edge $\mathsf{par}(v)v$ and instead, insert the 1-edge $\mathsf{par}(u)v$. All other edge labels remain unchanged. By construction, $\mathcal{X}[u,v]_{(T',\lambda')} = \{(u,v)\}$ and again, one easily checks that (T',λ') displays $\mathcal{X} \setminus \mathcal{X}[u,v]$ and therefore $\mathcal{X}$. Moreover, (T',λ') is by construction least resolved and therefore uniquely defined. Hence, it must locally look as in Fig. 50(1b, right). Reverting the local modifications in (T',λ') again yields the uniquely defined least resolved tree (T,λ) that explains $\mathcal{X}$ with $\mathcal{X}[u,v] = [u,v]$.

Case 2a. Suppose Δ_{uv} contains at least one triangle each of Types A and C but no triangles of Types B and D. Then $\mathcal{X}[u,v] \in \{u|v,(u,v),(v,u)\}$. We first assume that (T,λ) explains $u|v$. Then, as in Case 1a, u and v have to be siblings, none of them is a sink and no other 0-edge is incident to $\mathrm{lca}(u,v)$. Hence, (T,λ) locally looks again like Case 1a in Fig. 50. Local transformations of (T,λ) that are completely analogous to Case 1a can be applied to (T,λ) in order to obtain unique least resolved trees that explain $\mathcal{X}$ with $\mathcal{X}[u,v] = (u,v)$ and $\mathcal{X}[u,v] = (v,u)$, respectively (see Fig. 50(1a)). It is not hard to check that these transformations can be reversed by contraction of the edge vu.

Case 2b. If Δ_{uv} contains at least one triangle each of Types A and D but no triangles of Types B and C, then exactly the same arguments as in Cases 2a and 1a apply.

Case 2c. Suppose Δ_{uv} contains at least one triangle each of Types B and C but no triangles of Types A or D. Then $\mathcal{X}[u,v] \in \{(u,v),[u,v]\}$. Let us assume that (T,λ) explains $[u,v] \in \mathcal{X}$. As in Case 1b, the presence of triangles of Type B implies that v but not u is a sink. Arguing as in Case 1b shows that (T,λ) locally looks like Case 1b in Fig. 50. The local transformation to the least

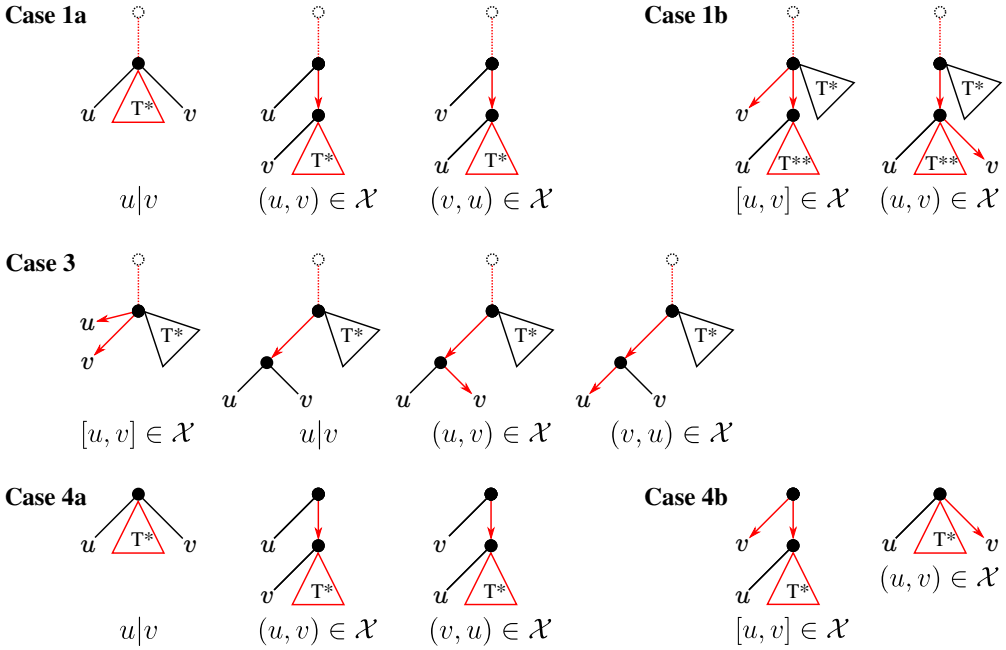


Fig. 50. Local modifications of the tree (T, λ) necessary to explain all possible choices of $\mathcal{X}[u, v]$ for different combination of triangles types in Δ_{uv}. Only the local environment around u and v is shown since the rest of the tree remains unchanged in all cases. Dashed lines indicate possible additional subtrees that are connected to the local situation by means of a 1-edge. The subtrees T^* and T^{**} (with red triangles) may be attached to an inner vertex (via 1-edges); their internal structure is irrelevant for the arguments in the proof.

resolved tree (T', λ') that explains $\mathcal{X}$ with $\mathcal{X}[u, v] = (u, v)$ can be performed as described in Case 1b, resulting in a tree (T', λ') that locally looks like Fig. 50(1b). The same arguments as in Case 1b can be applied to show that (T', λ') explains $\mathcal{X}$ and that there is a uniquely defined reverse transformation that converts (T', λ') into (T, λ).

Case 2d. If Δ_{uv} contains at least one triangle each of Types B and D but no triangles of Types A and C, exactly the same arguments as in Cases 2c and 1b apply.

Case 3. Suppose Δ_{uv} contains at least one triangle each of Types C and D but no triangles of Types A and B. Then $\mathcal{X}[u, v] \in \{[u, v], (u, v), (v, u), u|v\}$. Assume that (T, λ) explains $[u, v] \in \mathcal{X}$. This implies that both u and v are sinks of $\mathcal{X}$, i.e., $\mathsf{par}(u)u$ and $\mathsf{par}(v)v$ are both 1-edges. By symmetry, $(u, x) \in \mathcal{X}$ if and only if $(v, x) \in \mathcal{X}$ and $(x, u) \in \mathcal{X}$ if and only if $(x, v) \in \mathcal{X}$, respectively, holds for all $x \in L(T) \setminus \{u, v\}$.

We continue to show that u and v must be siblings. Assume for contradiction, they are not. Lemma 8.10(3) implies that there are distinct leaves $z, z' \in L(T) \setminus \{u, v\}$ such that $\mathsf{par}(u)z$ and $\mathsf{par}(v)z'$ are 0-edges. Hence, we have at least one of the cases $(u, z) \notin \mathcal{X}$ but $(v, z) \in \mathcal{X}$ or $(v, z') \notin \mathcal{X}$ but $(u, z') \in \mathcal{X}$. If $\mathsf{par}(u)$ and $\mathsf{par}(v)$ are incomparable in T, even both cases are true. However, we obtain a contradiction to "$(v, x) \in \mathcal{X}$ if and only if $(u, x) \in \mathcal{X}$". Thus u and v are siblings. Since $\mathsf{par}(u)u$ and $\mathsf{par}(v)v$ are both 1-edges, there must be a leaf

$y \in L(T) \setminus \{u, v\}$ such that the edge $\mathrm{lca}(u, v)y$ is a 0-edge by Lemma 8.10(3). We denote by T^* the subtree rooted at $\mathrm{lca}(u, v)$ without the leaves u and v.

Given (T, λ), a tree (T', λ') that displays $\mathcal{X}$ with $\mathcal{X}[u, v] = u|v$ is obtained by inserting an inner 1-edge ab such that a becomes the new root of T^* and $b = \mathrm{lca}_{T'}(u, v)$. The outer edges bu and bv are 0-edges; all other edge labels are retained as in (T, λ). The resulting tree locally looks as illustrated in Fig. 50(Case 3). Relabeling of edges in (T', λ') such that bv becomes a 1-edge yields the tree (T'', λ'') that explains $\mathcal{X}$ with $(u, v) \in \mathcal{X}$. Similarly, converting the edge bu of (T', λ') into a 1-edge yields the tree (T''', λ''') that explains $\mathcal{X}$ with $(v, u) \in \mathcal{X}$. The trees (T', λ'), (T'', λ''), and (T''', λ''') explain $\mathcal{X}$ with the corresponding choice of $\mathcal{X}[u, v]$ and are least resolved and thus unique. As in the previous cases, the reverse transformations are therefore also uniquely defined.

Case 4a. Suppose that all triangles in Δ_{uv} are of the form A. Then $\mathcal{X}[u, v] \in \{u|v, (u, v), (v, u)\}$. Let us assume that (T, λ) displays $u|v$. Then u and v are both sources, hence $\mathsf{par}(u)u$ and $\mathsf{par}(v)v$ are both 0-edges. Note that in contrast to Case 1a, there is no $x \in L(T) \setminus \{u, v\}$ with $(x, u) \in \mathcal{X}$ or $(x, v) \in \mathcal{X}$. This implies that u and v are both incident to the root ρ_T of (T, λ) and among all edges incident to the root, $\rho_T u$ and $\rho_T v$ are the only 0-edges. The tree (T, λ) explaining $\mathcal{X}[u, v] = u|v$ is shown Fig. 50(Case 4a). Note that the tree structure is very similar to Case 1a. Therefore, as in Case 1a, (T, λ) can be locally modified to a least resolved tree (T', λ') explaining $\mathcal{X}$ with $\mathcal{X}[u, v] = (u, v)$ by introducing the single 1-edge ab with $a = \mathsf{par}(u), b = \mathsf{par}(v)$. The vertex b becomes the root of T^*, where T^* is defined as in Case 1a (see Fig. 50(Case 4a)). We set $\lambda'(au) = 0$ and $\lambda'(bv) = 0$, while all other edge labels are retained.

Exchanging the roles of u and v in (T', λ') defines a least resolved tree (T'', λ'') that explains $\mathcal{X}[u, v] = (v, u)$. As in the previous cases, one easily verifies that all resulting trees are least resolved and explain $\mathcal{X}$ with the corresponding choice for $\mathcal{X}[u, v]$. Hence, the reverse transformations are also uniquely defined.

Case 4b. Suppose that Δ_{uv} contains only triangles of the form B. Hence, $\mathcal{X}[u, v] \in \{(u, v), [u, v]\}$. Let us first assume that the least resolved tree (T, λ) explains $[u, v] \in \mathcal{X}$. It immediately follows that v is a sink and u is not, hence $\lambda(\mathsf{par}(v)v) = 1$ and $\lambda(\mathsf{par}(u)u) = 0$. Moreover, we have $(v, u) \in \mathcal{X}$, thus $\mathsf{par}(v) \succ \mathsf{par}(u)$. Since for any $x \in L \setminus \{u, v\}$ it holds $(x, u) \notin \mathcal{X}$ and thus $\mathsf{par}(u) \succeq \mathrm{lca}(u, x)$, we have $\mathsf{par}(v) = \rho_T$ and $\deg(\rho_T) = 2$. Therefore (T, λ) locally looks as in Fig. 50(Case 4b). Note that the tree structure is very similar to Case 1b. Hence, similar as in Case 1b, (T, λ) can be locally modified to a least resolved tree (T', λ') that displays $(u, v) \in \mathcal{X}$ by contraction of $\mathsf{par}(v)\mathsf{par}(u)$ and keeping all other edge labels (see Fig. 50(Case 4b, right)). By the same argumentation as before, the reverse transformation is also uniquely defined.

Case 4c. Let us assume that all triangles in Δ_{uv} are of the form C, i.e., $\mathcal{X}[u, v] \in \{[u, v], u|v, (u, v), (v, u)\}$, and that (T, λ) explains $[u, v] \in \mathcal{X}$. As in Case 3, both u and v are sinks of $\mathcal{X}$, i.e., $\mathsf{par}(u)u$ and $\mathsf{par}(v)v$ are both 1-edges. Using the same symmetry argument as in Case 3, we conclude for any $x \in L(T) \setminus \{u, v\}$ that $(u, x) \in \mathcal{X}$ if and only if $(v, x) \in \mathcal{X}$, and $(x, u) \in \mathcal{X}$ if and only if $(x, v) \in \mathcal{X}$, respectively. Following the arguments laid out in Case 3, we conclude that (T, λ) locally looks as Case 3 of Fig. 50. Thus the local

transformations described above can be applied analogously in order to obtain least resolved trees that explain all possible $\mathcal{X}[u, v]$.

Case 4d. If Δ_{uv} contains only triangles of the form D, then we can apply the same construction as in Case 4a and 3 in order to conclude that $\mathcal{X}$ can be explained for all possible $\mathcal{X}[u, v]$. □

8.4 ALGORITHMIC CONSIDERATIONS

Summarizing our results, we present two different algorithms that are both able to recognize a Fitch relation and compute its unique least resolved tree. The first algorithm checks all induced triangles for forbidden subgraphs and, once recognized a Fitch relation, uses the set of informative triples as an input for the algorithm `BUILD`. Then it simply labels the edges of the resulting Aho tree in the correct way. This is a very intuitive way to check for Fitch relations and to construct the least resolved tree, which we will make precise first. We shall see that it is possible, however, to achieve a much better performance by using the fact that Fitch graphs are di-cographs. One can alternatively check for Fitch relations using properties of di-cographs and build the least resolved tree from the corresponding cotree. This can be achieved in linear time.

We have seen in the previous sections that every valid relation $\mathcal{X}$ is explained by a unique least resolved tree $(T_\mathcal{X}, \lambda_\mathcal{X})$, which, in turn, is identified by a set $\mathcal{R}(\mathcal{X})$ of informative triples due to Lemma 8.11. Lemma 3.1 therefore implies

$$T_\mathcal{X} = \text{Aho}(\mathcal{R}(\mathcal{X})) \tag{27}$$

It remains to construct the labeling function $\lambda_\mathcal{X}$ on $\text{Aho}(\mathcal{R}(\mathcal{X}))$.

Algorithm 8 `Label the Aho Tree`

Require: $T_\mathcal{X} = \text{Aho}(\mathcal{R}(\mathcal{X}))$
Ensure: Least resolved edge-labeled tree $(T_\mathcal{X}, \lambda_\mathcal{X})$ for $\mathcal{X}$

```
for all e = uv ∈ E(T) do
  if v ∉ L then
    λ_X(e) = 1
  else
    if (x, v) ∈ X for all x ∈ L \ {v} then
      λ_X(e) = 1
    else
      λ_X(e) = 0
```

Algorithm 8 has been implemented and tested by Anders [12].

Lemma 8.15. *Given the topology $T_\mathcal{X}$ of the unique least resolved tree explaining $\mathcal{X}$, Algorithm 8 computes its correct unique edge labeling $\lambda_\mathcal{X}$ in $\mathcal{O}(\max\{|\mathcal{X}|, |L|\})$ time.*

Proof. By Lemma 8.10, all inner edges e of $\text{Aho}(\mathcal{R}(\mathcal{X}))$ must be labeled $\lambda(e) = 1$ since otherwise they could be contracted and hence, the tree would not be least resolved. Now consider an edge $e = uv$ leading to a leaf $v \in L$. If

$(x, v) \notin \mathcal{X}$ for some $x \in L \setminus \{v\}$, then $\lambda(e) = 0$. Conversely, if $\lambda(e) = 0$, then $(x, v) \notin \mathcal{X}$ for every leaf below the siblings of u. At least one such leaf x exists in a phylogenetic tree. Hence, an outer edge is labeled $\lambda(e) = 1$ if and only if $(x, v) \in \mathcal{X}$ for all $x \in L \setminus \{v\}$.

For the time complexity note that the labeling Algorithm 8 requires $\mathcal{O}(|E(T_{\mathcal{X}})|)$ operations to label the inner edges. In order to label the $|L|$ outer edges uv we have to determine the degree of vertex v in $\mathcal{X}$, that is $\deg(v) = 1$ implies that uv is an outer edge, which requires $\mathcal{O}(\max\{|\mathcal{X}|, |L|\})$ operations. Since $|E(T_{\mathcal{X}})|$ is bounded by $\mathcal{O}(|L|)$, the total running time of the labeling step is bounded by $\mathcal{O}(\max\{|\mathcal{X}|, |L|\})$. □

A tree explaining a given Fitch relation can be obtained by the following procedure: First, we check whether $\mathcal{X}$ is a Fitch relation. This can be achieved in $\mathcal{O}(|L|^3)$ by checking validity of the $\binom{L}{3}$ induced triangles. If $\mathcal{X} \subset L \times L$ is a Fitch relation, then $\mathcal{R}(\mathcal{X})$ can be constructed within $\mathcal{O}(|L|^3)$ time. For a given the set of triples $\mathcal{R}(\mathcal{X})$, the original approach to check whether $\mathcal{R}(\mathcal{X})$ is consistent (in which case $\mathrm{Aho}(\mathcal{R}(\mathcal{X}))$ is returned) or not, has time complexity $\mathcal{O}(|\mathcal{R}(\mathcal{X})||L|)$ [3]. However, various further practical implementations have been described [104, 118, 110, 118] that improve the asymptotic performance. Constructing $\mathrm{Aho}(\mathcal{R}(\mathcal{X}))$ and using Algorithm 8 to obtain the edge labels, it is therefore possible to recognize a Fitch relation $\mathcal{X}$ and to compute its respective (least resolved) tree (T, λ) in $\mathcal{O}(|L|^4)$.

It is possible to improve the algorithms to recognize Fitch relations $\mathcal{X}$ and compute its least resolved tree (T, λ) using di-cotrees (cf. Chapter 3 for a definition). As discussed in Section 8.1, any di-cograph that does not contain the invalid triangles F_1, F_5, or F_8 is a Fitch graph.

Lemma 8.16. *Let $\vec{G}$ be a di-cograph and $(T', \vec{t})$ its corresponding di-cotree. A di-cograph contains the triangle F_1, F_5, or F_8 as an induced subgraph if and only if there are two vertices $v, w \in V^0(T')$ with $v \succ_{T'} w$ such that either (i) $\vec{t}(v) = 0 \neq \vec{t}(w)$, or (ii) $\vec{t}(v) = \vec{1}$, $\vec{t}(w) = 1$ and w is located in some subtree (rooted at a child of v) that is different from the subtree rooted at the right-most child of v.*

Proof. Consider first the triangles F_1 and F_5 with vertices x, y, z and edge set $E(F_1) = \{(x, y)\}$ and $E(F_5) = \{(x, y), (y, x)\}$. Equivalently, we have $\vec{t}(\mathrm{lca}_{T'}(x, y)) \in \{1, \vec{1}\}$, $\vec{t}(\mathrm{lca}_{T'}(x, y, z)) = 0$ and $v = \mathrm{lca}_{T'}(x, y, z) \succ_{T'} w = \mathrm{lca}_{T'}(x, y)$.

Now let F_8 have vertices x, y, z and edge set $E(F_8) = \{(x, y), (y, x), (x, z), (y, z)\}$. Equivalently, we have $\vec{t}(\mathrm{lca}_{T'}(x, y)) = 1$, $\vec{t}(\mathrm{lca}_{T'}(x, y, z)) = \vec{1}$ and $v = \mathrm{lca}_{T'}(x, y, z) \succ_T w = \mathrm{lca}_{T'}(x, y)$. In particular, x and y must be placed left from z in T' and therefore, w must be located in some subtree different from the subtree rooted at the right-most child of $v = \mathrm{lca}_{T'}(x, y, z)$. □

Corollary 8.1. *Let $\mathcal{X}$ be a Fitch graph and $(T', \vec{t})$ its corresponding di-cotree. If $\mathcal{X}$ contains an edge, then it is connected. Moreover, any vertex $x \prec v$ for which $\vec{t}(v) = 0$ must be a leaf of T'.*

Proof. If a Fitch graph $\mathcal{X}$ contains an edge, then its di-cotree contains an inner vertex labeled 1 or $\vec{1}$. If $\mathcal{X}$ is disconnected, then the root of the cotree must be labeled 0 and Lemma 8.16 implies that $\mathcal{X}$ is not a Fitch graph. Thus the root must be labeled either 1 or $\vec{1}$, which implies that $\mathcal{X}$ is connected.

Now assume that $(T', \vec{t})$ contains a vertex v with $\vec{t}(v) = 0$. Let $x \prec v$ with $vx \in E(T')$ and assume, for contradiction, that x is an inner vertex. By the definition of di-cotrees, $\vec{t}(v) = 0 \neq \vec{t}(x)$. Lemma 8.16 and Thm. 8.2 imply that $\mathcal{X}$ is not a Fitch graph; a contradiction. □

Verifying whether a graph $\vec{G}$ is a di-cograph or not can be achieved in $\mathcal{O}(|V(\vec{G})| + |E(\vec{G})|)$ time, see [157, 100] for further details. To verify that a given di-cograph G does not contain F_1, F_5, and F_8 as an induced subgraph, we apply the classical Breadth-first search (BFS) [36] on its di-cotree $(T', \vec{t})$ starting with the root and check whether there are invalid combinations of vertex labels in $(T', \vec{t})$ according to Lemma 8.16. Note that $L(T') = V(\vec{G})$ and $|V^0(T')| \leq |L(T')| - 1$. Thus the BFS-method runs in $\mathcal{O}(|V(T')|) = \mathcal{O}(|V(\vec{G})|)$ time. Therefore recognition of Fitch graphs or, equivalently, Fitch relations can be achieved within $\mathcal{O}(|V(\vec{G})| + |E(\vec{G})|)$ time.

We now show how to obtain a tree (T, λ) that explains a Fitch relation $\mathcal{X}$ from its di-cotree representation $(T', \vec{t})$. To this end we need to translate the (ordered) di-cotree with vertex labels "0", "1" and "$\vec{1}$" to an unordered tree with edge labels "1" and "0", summarized next and called `cotree2fitchtree`:

For all $x \in V^0(T')$, if

- $\vec{t}(x) = 1$ (resp. 0), then set for each child y of x the label $\lambda(xy) = 1$ (resp. 0), and else,
- $\vec{t}(x) = \vec{1}$, then we can assume w.l.o.g. that the children of x are ordered $x_1, \ldots, x_k$, $k \geq 2$ from left to right. Now, replace the subtree of T' with vertices x and $x_1, \ldots, x_k$ by the caterpillar $C(x_1, \ldots, x_k) := (x_1(x_2(\ldots(x_{k-1}, x_k)\ldots)$ *(in Newick notation)* that is rooted at x. Set the label λ of all inner edges of $C(x_1, \ldots, x_k)$ and the outer-edge incident to x_k to "1" and the labels of all other (outer) edges of $C(x_1, \ldots, x_k)$ to "0". Note that outer edges of $C(x_1, \ldots, x_k)$ may be inner edges in (T, λ).

Finally, remove all vertex labels and ignore the ordering of the vertices to obtain the tree (T, λ).

For an example of `cotree2fitchtree` see Fig. 51.

Lemma 8.17. *The procedure* `cotree2fitchtree` *transforms the di-cotree* $(T', \vec{t})$ *of a Fitch relation* $\mathcal{X}$ *into a tree* (T, λ) *that explains* $\mathcal{X}$ *in* $\mathcal{O}(|V(T')|)$ *time.*

Proof. Let $(T', \vec{t})$ be the di-cotree of the Fitch relation $\mathcal{X}$ and (T, λ) the tree resulting from `cotree2fitchtree`. Since all inner vertices of $(T', \vec{t})$ are labeled, each edge of (T, λ) receives a label "0" or "1" by construction. It needs to be verified that (T, λ) explains $\mathcal{X}$.

Assume $(x, y), (y, x) \in \mathcal{X}$. Hence, $\vec{t}(\mathrm{lca}_{T'}(x, y)) = 1$. By construction, the edges incident to the children of $v = \mathrm{lca}_{T'}(x, y)$ are labeled "1". Hence, both

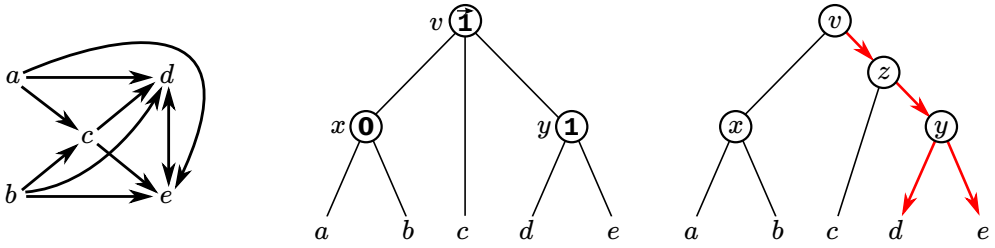


Fig. 51. Application of `cotree2fitchtree`: A Fitch relation $\mathcal{X}$ (left), its di-cotree $(T', \vec{t})$ (middle), and an edge-labeled tree (T, λ) that explains $\mathcal{X}$ (right) is shown. The tree (T, λ) is obtained from $(T', \vec{t})$ by replacing the subtree with vertices v, x, y and c by the caterpillar $(x(c, y))$ rooted at v and adding the edge labels as described in the procedure `cotree2fitchtree`. By Lemma 8.16, $\vec{t}(x) = 0$ for all inner vertices x in the subtrees left from the subtree rooted a the right-most child y of v. Note that (T, λ) is not least resolved w.r.t. $\mathcal{X}$. Nevertheless, Thm. 8.1 implies that (T, λ) displays least resolved tree for $\mathcal{X}$. Here, the least resolved tree can be obtained from (T, λ) by ex-contracting the edges vx and zy.

paths in (T, λ) from $\mathrm{lca}_T(x, y) = v$ to x and to y contain 1-edges. Thus (T, λ) explains all symmetric pairs in $\mathcal{X}$.

Assume $(x, y), (y, x) \notin \mathcal{X}$ and let $z = \mathrm{lca}_{T'}(x, y)$. Hence, $\vec{t}(z) = 0$. Cor. 8.1 implies that zx and zy are outer edges in T' that are, by construction, labeled "0" in (T, λ). As a consequence, the path from x to y in (T, λ) contains only 0-edges, which implies that (T, λ) also explains that all pairs $(x, y), (y, x)$ that are not contained in $\mathcal{X}$.

Assume $(x, y) \in \mathcal{X}$ and $(y, x) \notin \mathcal{X}$. Hence, $\vec{t}(\mathrm{lca}_{T'}(x, y)) = \vec{1}$ and x is left from y in T'. Let v_i and v_j be children of $\mathrm{lca}_{T'}(x, y)$ with $v_i \succeq x$ and $v_j \succeq y$. Since x is left from y, also v_i is left from v_j in T'. Note, v_i and v_j are now part of the inserted caterpillar $C(\mathsf{child}(\mathrm{lca}_{T'}(x, y)))$ in (T, λ). Therefore $\mathrm{lca}_T(x, y)$ must be an inner vertex of this caterpillar. By construction, the path from $\mathrm{lca}_T(x, y)$ to $v_j \succeq y$ contains a 1-edge and thus, $(x, y) \in \mathcal{X}$. It remains to show that the path from $\mathrm{lca}_T(x, y)$ to x contains only 0-edges so that $(y, x) \notin \mathcal{X}$. Note that the vertex v_i is a child of $\mathrm{lca}_T(x, y)$ in T and the edge $\mathrm{lca}_T(x, y)v_i$ is labeled "0". Thus, if $v_i = x$, we are done. Assume that $v_i \neq x$ and hence, v_i is an inner vertex of T'. By the definition of di-cotrees, we have $\vec{t}(\mathrm{lca}_{T'}(x, y)) = \vec{1} \neq \vec{t}(v_i)$. Since v_i is left from v_j in T', we can apply Lemma 8.16 and conclude that $\vec{t}(v_i) \neq 1$. Hence, there is only one possibility left, namely $\vec{t}(v_i) = 0$. Cor. 8.1 implies that $v_i x$ must must be an outer edge in $(T', \vec{t})$ that – by construction – is labeled "0" in (T, λ). Hence, the path from $\mathrm{lca}_T(x, y)$ to x contains only 0-edges and therefore, $(y, x) \notin \mathcal{X}$.

For the running time, observe that the edge label in each step of `cotree2fitchtree` for vertices v with $\vec{t}(v) \in \{0, 1\}$ can be computed in $\mathcal{O}(\deg_{T'}(v))$ time. Moreover, if $\vec{t}(v) = \vec{1}$ for some vertex v in $(T', \vec{t})$, we have to replace the subtree induced by v and its children $v_1, \ldots, v_k$ (ordered from left to right) in $(T', \vec{t})$, by the edge-labeled caterpillar $C(v_1, \ldots, v_k)$. This task can also be performed in $\mathcal{O}(\deg_{T'}(v))$ time. Since each step in `cotree2fitchtree` can be done in $\mathcal{O}(\deg_{T'}(v))$ time and $\sum_{v \in V^0(T')} \deg_{T'}(v)) \leq 2|E(T')| < 2|V(T')|$, this implies a total time requirement of $\mathcal{O}(|V(T')|)$. □

Let (T, λ) be the tree that explains $\mathcal{X}$ as constructed with `cotree2fitchtree` from the respective di-cotree $(T', \vec{t})$. Thm. 8.1 implies that (T, λ) displays the least resolved tree for $\mathcal{X}$. Thus we can utilize Lemma 8.10 and *ex*-contract all irrelevant edges and all inner 0-edges in (T, λ) in order to obtain the least resolved tree for $\mathcal{X}$. The latter can be done in $\mathcal{O}(|V(T)|)$ time. Taking the latter results together with the observation that $|V(T)| \geq |V(T')|$, we obtain the following

Theorem 8.3. *Verifying whether an irreflexive relation $\mathcal{X} \subseteq L \times L$ is a Fitch relation or not, can be a achieved in $\mathcal{O}(|L| + |\mathcal{X}|)$ time. Its unique least resolved edge-labeled tree $(T_\mathcal{X}, \lambda_\mathcal{X})$ can be computed in $\mathcal{O}(|V(T_\mathcal{X})|) = \mathcal{O}(|L|)$ time, given the di-cotree of $\mathcal{X}$.*

The fact that Fitch graphs form a heritable family (cf. Lemma 8.2) has far-reaching consequences for computational problems such as:

Problem 8.1 (Fitch graph (i, j, k)-modification)**.**

Given:	*A directed graph $\vec{G} = (L, \vec{E})$ and non-negative integers i, j, k.*
Question:	*Are there subsets $L' \subseteq L$, $\vec{E}' \subseteq \vec{E}$ and $\vec{E}'' \subseteq (L \times L) \setminus \vec{E}$ with $\lvert L' \rvert \leq i$, $\lvert \vec{E}' \rvert \leq j$ and $\lvert \vec{E}'' \rvert \leq k$ such that $\vec{G} - L' - \vec{E}' + \vec{E}''$ is a Fitch graph?*

As a consequence of the results established in [235, 147], the corresponding vertex deletion problem (i.e., the restriction in which only edges are deleted that are incident to removed vertices) is NP-complete. Here, only $|L'| \leq i$ is specified as part of the problem. NP-completeness of other types of Fitch graph modification problems is still open. Most likely they are NP-complete as well. Here, we show that the Fitch graph (i, j, k)-modification problem for any $i, j, k \geq 0$ is fixed-parameter tractable (FPT), see [170] for more details regarding FPT.

Since Fitch graphs have a characterization in terms of a *finite* set of forbidden subgraphs, we obtain as an immediate consequence of Thm. 1 in [29] and Thm. 8.3:

Corollary 8.2. *If a directed graph $\vec{G}$ with vertex set L is not a Fitch graph, then a forbidden subgraph can be determined in $\mathcal{O}(n^2 + nm)$ time, where $n = |L|$ and $m = |E(\vec{G})|$.*

In order to obtain an FPT-algorithm, we reuse the results as provided in the proof of Thm. 1 and 2 in [29]. Consider the following simple procedure:

1. Find a forbidden triangle F in $\vec{G}$ (in $\mathcal{O}(n^2 + nm)$ time).
2. Modify $\vec{G}$ by either deleting an edge or a vertex from F, or adding an edge to F.

Clearly, the Fitch graph (i, j, k)-modification problem is solved by repeating these two steps until one either obtains a Fitch graph or the "allowance" of i vertex deletions, j edge deletions, and k edge additions is exhausted.

To estimate the time complexity of this procedure, we note that the forbidden subgraphs are triangles, i.e., the number of vertices in the forbidden subgraphs

is always $N = 3$. Moreover, there are are at most $2\binom{N}{2} = 6$ different ways to add an edge to or delete an edge from F, and at most $N = 3$ different ways to delete a vertex from F. Thus one can enumerate all possible ways to delete $\leq i$ vertices and $\leq j$ edges, and to add $\leq k$ edges in at most $6^{j+k}3^i$ repetitions of Step 1 and 2. Each step requires $\mathcal{O}(n^2 + nm)$ time to find a forbidden triangle. Therefore we obtain

Corollary 8.3. *Fitch graph (i, j, k)-modification is fixed-parameter tractable and can be solved in $\mathcal{O}(6^{j+k}3^i(|L|^2 + |L||E(\vec{G})|))$ for a given directed graph $\vec{G}$ with vertex set L.*

8.5 THE SYMMETRIC FITCH RELATION

Motivated by the fact that the direction of HGT events cannot always be unambiguously inferred from sequence data, it is natural to consider also the symmetrized version of the Fitch relation, i.e., to interpret the undirected edge xy as a xenologous pair whenever the evolutionary history separated x and y by at least one horizontal transfer event. In mathematical terms, this idea is captured by

Definition 8.10. *Let T be a rooted tree with leaf set L and let $\lambda : E(T) \to \{0, 1\}$. Then the* undirected Fitch graph $\mathcal{X}^{\mathrm{sym}}$ *explained by (T, λ) has vertex set L and edges $xy \in E(\mathcal{X}^{\mathrm{sym}})$ if and only if the (unique) path from x to y in T contains at least one edge e with $\lambda(e) = 1$. A graph $\mathcal{X}^{\mathrm{sym}}$ is an undirected Fitch graph if and only if it is explained in this manner by some edge-labeled rooted tree (T, λ).*

Undirected Fitch graphs are closely related to their directed counterparts. Since the path P connecting two leaves x and y in an edge-labeled rooted tree (T, λ) is unique and contains their last common ancestor $\mathrm{lca}(x, y)$, there is a 1-edge along P if and only if there is a 1-edge on the path between x and $\mathrm{lca}(x, y)$ or between $\mathrm{lca}(x, y)$ and y. The undirected Fitch graph is therefore the underlying undirected graph of the directed Fitch graph, i.e., it is obtained from the directed version by ignoring the direction of the arcs.

The undirected Fitch graphs form a heritable family, i.e., if $\mathcal{X}^{sym}$ is an undirected Fitch graph, so are all its induced subgraphs. This is an immediate consequence of the fact that directed Fitch graphs are a heritable family of digraphs. The fact can also be obtained directly by considering the restriction of T to a subset of leaves. This obviously does not affect the paths or their labeling between the remaining vertices.

Clearly, $\mathcal{X}^{sym}$ does not depend on which of the non-leaf vertices in T is the root. Furthermore, a vertex v with only two neighbors and its two incident edges e' and e'' can be replaced by a single edge e. The new edge is labeled $\lambda(e) = 0$ if both $\lambda(e') = \lambda(e'') = 0$, and $\lambda(e) = 1$ otherwise. These operations do not affect the undirected Fitch graph. Hence, we can replace the rooted tree T by an unrooted tree in Def. 8.10 and assume that all non-leaf edges have at least degree 3. To avoid trivial cases we assume throughout that T has at least two leaves and hence, an undirected Fitch graph has at least two vertices.

Lemma 8.18. *If $\mathcal{X}^{\mathrm{sym}}$ is an undirected Fitch graph, then $\mathcal{X}^{\mathrm{sym}}$ does not contain $K_1 \cup K_2$ as an induced subgraph. In particular every undirected Fitch graph is a complete multipartite graph.*

Proof. There is a single unrooted tree with three leaves, namely the star S_3, which admits four non-isomorphic $\{0,1\}$-edge labelings defined by the number N of 1-edges. The undirected Fitch graphs $\mathcal{X}_N^{sym}$ are easily obtained. In the absence of 1-edges, $\mathcal{X}_0^{sym} = \overline{K_3}$ is edge-less. For $N = 2$ and $N = 3$ there is a 1-edge along the path between any two leaves, i.e., $\mathcal{X}_2^{sym} = \mathcal{X}_3^{sym} = K_3$. For $N = 1$ one leaf is connected to the other two by a path in S_3 containing a 1-edge; the path between the other two leaves consists of two 0-edges, hence $\mathcal{X}_1^{sym}$ corresponds to the induced path of length two. Hence, only three of the four possible undirected graphs on three vertices can be realized, while $K_1 \cup K_2$ is not an undirected Fitch graph. By heredity, $K_1 \cup K_2$ is therefore a forbidden induced subgraph for the class of undirected Fitch graphs. Finally, it is well known that the class of graphs that do not contain $K_1 \cup K_2$ as an induced subgraph are exactly the complete multipartite graphs, see e.g. [242]. □

Note that the first part of Lemma 8.18 can also be obtained from the eight directed Fitch graphs on three vertices, using the fact that an undirected Fitch graph is the underlying (undirected) graph of a directed Fitch graph.

In order to show that forbidding $K_1 \cup K_2$ is also sufficient, we explicitly construct the edge-labeled trees necessary to explain complete multipartite graphs. To this end, recall that each complete multipartite graph $K_{n_1,\dots,n_k}$ is determined by its independent sets $V_1, \dots, V_k$ with $|V_i| = n_i$ for $1 \le i \le k$, where $xy \in E(K_{n_1,\dots,n_k})$ if and only if $x \in V_i$ and $y \in V_j$ with $i \neq j$. In particular, therefore, $K_{n_1,\dots,n_k}$ with at least two vertices is connected if and only if $k \ge 2$. The complete 1-partite graphs are the edge-less graphs $\overline{K}_n$.

Since $K_1 \cup K_2$ is an induced subgraph of the path on four vertices P_4, any graph G that does not contain $K_1 \cup K_2$ as an induced subgraph must be P_4-free, i.e., a cograph [38]. The cotrees of connected multipartite graphs have a particularly simple shape, illustrated without the vertex labels in Fig. 52. The cotree has a root labeled "1" and all inner vertices labeled "0". Here we do not need the connection between cographs and their cotrees, however. Therefore we introduce these trees together with an edge labeling that is useful for our purposes in the following

Definition 8.11. *For $k = 1$, $T[n]$ is the star tree S_n with n leaves. For $k \ge 2$, the tree $T[n_1, \dots, n_k]$ has a root ρ with k children c_i, $1 \le i \le k$. The vertex c_i is a leaf if $|V_i| = n_i = 1$ and has exactly n_i children that are leaves if $|V_i| = n_i \ge 2$.*
For $k = 1$, all edges e of $T[n]$ are labeled $\lambda^*(e) = 0$. For $k \ge 2$, we set $\lambda^*(\rho c_i) = 1$ for $1 \le i \le k$ and $\lambda^*(e) = 0$ for all edges not incident to the root.

Now we can prove our main result:

Theorem 8.4. *A graph G is an undirected Fitch graph if and only if it is a complete multipartite graph. In particular, $K_{n_1,\dots,n_k}$ is explained by $(T[n_1, \dots, n_k], \lambda^*)$.*

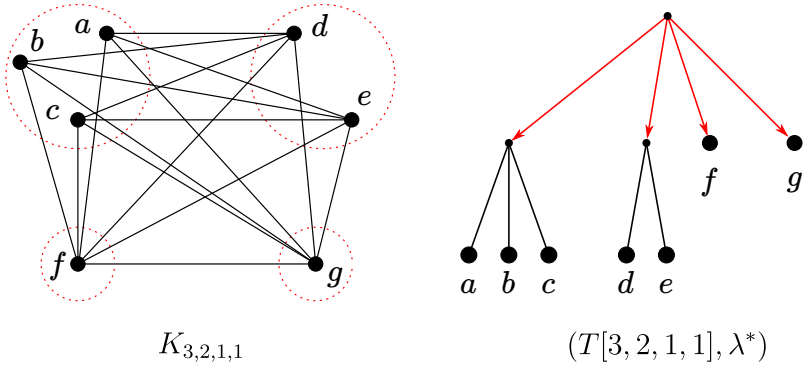


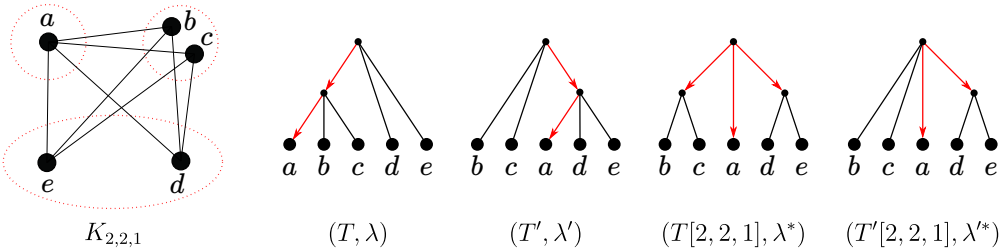
Fig. 52. The complete multipartite graph $K_{3,2,1,1}$ is the Fitch graph explained by the tree $T[3,2,1,1]$ with edge labeling λ^*, where 0-edges and 1-edges are drawn in black and red, respectively.

Proof. Lemma 8.18 implies that an undirected Fitch graph is a complete multipartite graph. To show the converse, we fix an arbitrary complete multipartite graph $G = K_{n_1,\dots,n_k}$ and find an edge-labeled rooted tree (T,λ^*) that explains G.
For $k = 1$ it is trivial that $(T[n],\lambda^*)$ explains $\overline{K_n}$.
For $k \geq 2$ consider the tree $T[n_1,\dots n_k]$ with edge labeling λ^* and let $\mathcal{X}^{sym}$ be the corresponding undirected Fitch graph. The leaf set of $T[n_1,\dots n_k]$ is partitioned into exactly k subsets $L_1,\dots,L_k$ defined by (a) singletons adjacent to the root and (b) subsets comprising at least two leaves adjacent to the same child c_i of the root. Furthermore, we can order the leaf sets so that $|L_i| = n_i$. By construction, all vertices within a leaf set L_i are connected by a path that does not run through the root and thus, contains only 0-edges if $|L_i| > 1$ and no edge, otherwise. The L_i are independent sets in $\mathcal{X}^{sym}$. On the other hand any two leaves $x \in L_i$ and $y \in L_j$ with $i \neq j$ are connected only by path through the root, which contains two 1-edges. Thus x and y are connected by an edge in $\mathcal{X}^{sym}$ and therefore, $\mathcal{X}^{sym}$ is a complete multipartite graph of the form $K_{|L_1|,\dots,|L_k|} = K_{n_1,\dots,n_k}$. Since $K_{n_1,\dots,n_k}$ is explained by $(T[n_1,\dots,n_k],\lambda^*)$ for all $n_i \geq 1$, $k \geq 2$ and $\overline{K_n}$ is explained by $(T[n],\lambda^*)$, we conclude that every complete multipartite graph is an undirected Fitch graph. □

The converse of Lemma 8.18 does not follow in a straightforward manner from the characterization of directed Fitch graphs. It is possible to make use of the connection between directed Fitch graphs and di-cographs to obtain the trees of Def. 8.11. This line of reasoning, however, is neither shorter nor simpler than the direct, elementary proof given above.

Complete multipartite graphs $G = (V,E)$ obviously can be recognized in $O(|V|^2)$ time (e.g. by checking that its complement is a disjoint union of complete graphs), and even in $O(|V| + |E|)$ time by explicitly constructing its modular decomposition tree [158]. Given the tree $T[n_1,\dots,n_k]$, the canonical edge labeling λ^* is then assigned in $O(|V|)$ time.

A tree (T,λ) that explains an undirected Fitch graph $\mathcal{X}^{sym}$ is *minimal* if it has the smallest number of vertices among all trees that explain $\mathcal{X}^{sym}$. In



$K_{2,2,1}$ (T,λ) (T',λ') $(T[2,2,1],\lambda^*)$ $(T'[2,2,1],\lambda'^*)$

Fig. 53. The non-isomorphic trees (T,λ), (T',λ'), $(T[2,2,1],\lambda^*)$, and $(T'[2,2,1],\lambda'^*)$ all explain the same complete multipartite graph $K_{2,2,1}$. Three of these trees have the smallest possible number (7) of vertices and thus are minimal. These can be obtained from the tree $(T[2,2,1],\lambda^*)$ specified in Def. 8.10 by contraction of one of its inner 1-edges and possibly re-routing the resulting tree.

this case, (T,λ) is also *least resolved*, i.e., the contraction of any edge in (T,λ) results in a tree that does not explain $\mathcal{X}^{sym}$. Not surprisingly, the tree $(T[n_1,\dots,n_k],\lambda^*)$ is almost minimal in most, and minimal in some cases: Since the vertices of the Fitch graph must correspond to leaves of the tree, $(T[n_1,\dots,n_k],\lambda^*)$ is necessarily minimal whenever it is a star, i.e., for $(T[n],\lambda^*)$ and $(T[1,\dots,1],\lambda^*)$. In all other cases, its only potentially "superfluous" part is its root. Indeed, exactly one of the edges connecting the root with a non-leaf neighbor can be contracted without changing the corresponding Fitch graph. It is clear that this graph is minimal: The leaf sets L_i must be leaves of an induced subtree without an intervening 1-edge. Having all vertices of L_i adjacent to the same vertex is obviously the minimal choice. Since the L_i must be separated from all other leaves by a 1-edge, at least one incident edge of c_i must be a 1-edge. Removing all leaves incident to a 0-edge results in a tree with at least k vertices that must contain at least $k-1$ 1-edges because every path between leaves in this tree must contain a 1-edge. The contraction of exactly one of the k 1-edges incident to the root ρ in $T[n_1,\dots,n_k]$ indeed already yields a minimal tree. In general, the minimal trees are not unique, see Fig. 53.

It may be worth noting that $K_{n_1,\dots,n_k}$ can also be explained by binary trees. To see this, we convert a tree $(T[n_1,\dots,n_k],\lambda^*)$ into a binary tree in two simple steps. First, each group of $n_i > 1$ leaves with a common parent is replaced by an arbitrary binary tree with the same leaf set and all edges labeled 0. Second, the star consisting of the root and all its children C is replaced by an arbitrary rooted binary tree with leaf set C and all edges labeled 1. It is obvious that neither of the operations affects the graph that is explained.

8.6 SUMMARY

The first part of this chapter formalized Fitch's concept of xenology in the form of a not necessarily symmetric binary relation $\mathcal{X}$ such that $(x,y) \in \mathcal{X}$ if and only the lineage from $\mathrm{lca}(x,y)$ to y was horizontally transferred at least once. The main result is a complete characterization of such relations in terms of forbidden induced subgraphs and a complete characterization of the minimally resolved trees explaining such relations. These Fitch trees

represent the complete information on the gene tree that is "recorded" by the horizontal transfer events alone. Moreover, polynomial-time algorithms have been devised to compute Fitch trees from Fitch relations.
For the undirected Fitch relation it has been shown that a graph is an undirected Fitch graph if and only if it is a complete multipartite graph (cf. Thm. 8.4).

9 CONCLUSION

The focus of this work laid on a thorough analysis of the best match and the reciprocal best match relation as well as their connection to the orthology relation in the case of duplication/loss scenarios. Another main point was the problem how to retrieve best matches from sequence data. Moreover, the impact of horizontal gene transfer in the context of correct orthology assignment was empirically investigated and a complete characterization of the corresponding xenology relation was given.

First, this thesis provided two characterizations of 2-colored best match graphs: One in terms of three simple conditions on the out-neighborhoods and the other via informative triples that can be directly extracted from the input graph. General BMGs with more than two colors were then characterized via the set of their induced 2-BMGs. In both cases, polynomial time algorithms for the recognition of BMGs and tree reconstruction were given. Moreover, it was shown that for any BMG $(\vec{G}, \sigma)$ there exists a unique *least resolved* tree, i.e., a tree with the lowest possible resolution that explains $(\vec{G}, \sigma)$.

A characterization of RBMGs was given in Chapter 5, where it has been demonstrated that 3-RBMGs fall into three distinct classes, one of which has cograph structure. Similarly to BMGs, a characterization of RBMGs with more than three species was given in terms of the induced 3-RBMGs. Reciprocal best match graphs - in contrast to best match graphs - have a surprisingly complicated structure which makes their recognition quite difficult. Although 3-RBMGs can be recognized in polynomial time, it remains an open question whether the problem of recognizing RBMGs with more than three colors can be solved in polynomial time. It is also unknown whether the information contained in triples derived from three-colored connected components is sufficient, even if this may not lead to a polynomial time recognition algorithm.

As shown in Chapter 6, the true orthology relation is a subgraph of the reciprocal best match graph in the absence of HGT events, i.e., the reciprocal best match graph only contains false positive orthology assignments. Moreover, a certain pattern, called *good quartets*, could be identified in the underlying best match graph that can be used in order to find false positive edges in the RBMG. The empirical simulations presented here revealed that good quartets identify almost all false positive edges in the absence of HGT. However, in the presence of HGT, the RBMG and the true orthology graph Θ heavily deviate and the removal of edges identified by good quartets introduces false negative edges.

Chapter 7 demonstrated that local information in form of a subset of quartets is sufficient for the estimation of best matches, instead of reconstructing whole gene trees. Furthermore, the theoretical results in this chapter give some guarantees for obtaining the correct best matches from quartets and highlight some limitations that cannot be overcome with certainty as long as only dis-

tance data is available. In particular, correct best match estimates crucially depend on the identification of suitable outgroups.

Finally, the directed Fitch relation was characterized by forbidden subgraphs on three leaves and a polynomial time algorithm for the reconstruction of the corresponding unique least resolved tree was provided. Moreover, the undirected Fitch relation turned out to be a complete multipartite graph.

In the case of BMGs, the existence of a unique least resolved tree implicates that this tree must be a homeomorphic image of the gene tree that explains the true (but usually unknown) evolutionary scenario. In contrast, there exists in general no unique least resolved tree for RBMGs. By focusing on reciprocal best matches and ignoring the directed best matches, important information about the true evolutionary history of a given gene family is thus ignored. This circumstance suggests to not only incorporate reciprocal best matches but also non-symmetric best matches into orthology detection methods. This is strongly supported by the simulation results in Chapter 6, where the removal of false positive edges identified by good quartets drastically reduces the number of induced P_4s. Furthermore, this observation suggests to consider *hc*-cograph editing with a given best match relation. It seems most likely that orthology detection pipelines could be substantially improved by first inserting BMG and RBMG editing and then removing all good P_4s, followed by a variant of cograph editing that respects the *hc*-cograph structure. While cograph editing is an NP-complete problem in general [150], the complexity of the colored version, i.e., editing a properly colored graph to the nearest *hc*-cograph remains unknown. However, it seems likely that the additional knowledge of the directed edges in the BMG makes the problem tractable since it already implies a unique least resolved tree that captures much of the cograph structure.
Cograph editing would be fully content with *hc*-cographs, i.e., RBMGs that are cographs. These are not necessarily "biologically feasible" in the sense that they can be reconciled with a species tree. It will therefore also be of interest to consider the problem of editing an *hc*-cograph to another *hc*-cograph that is reconcilable with some or a given species tree – a problem that has already been considered for orthology relations [141, 138]. Since the obstructions are conflicting triples with a speciation at their top node, the offending data represents conflicting orthology assignments. Therefore, it seems natural to ask for a maximal induced sub-*hc*-cograph that implies a consistent triple set, instead of phrasing the problem as an arbitrary editing problem. If it is indeed true that triples necessarily displayed by the species tree can be extracted directly from the (R)BMG, it will be of practical use to consider the corresponding edge deletion problem for (R)BMGs. In particular, it would be interesting to know whether the latter problem is the same as asking for the maximal consistent subset of triples implied by the c(R)BMG or co-BMG.
BMG, RBMG, and *hc*-cograph editing is the subject of ongoing research (see e.g. [102]).

Furthermore, Chapter 6 showed that good quartets can occur in different contexts, where some of them refer to false positive orthologs without HGT involved and others correspond to HGT events. These contexts need to be treated differently in the editing problem. From a more theoretical point of

view, the empirical findings in this chapter beg two questions: (1) Are there *local* features in the (R)BMG that make it possible to unambiguously identify HGT at least in some cases? (2) What kind of additional information can be integrated to distinguish good quartets that arise from duplication/loss events from those that are introduced by HGT, i.e., how can it be decided if good quartets can be safely removed or should be "repaired" in a different manner? In particular, can complete or partial information on the Fitch relation be integrated, e.g. to provide additional constraints on the trees explaining a given RBMG? Moreover, the Fitch relation corresponds to a subclass of directed cographs, which are also connected to a generalization of orthology relations that incorporate HGT events [100]. It seems therefore worthwhile to explore whether there is a direct connection between BMGs and directed cographs, possibly for those BMGs whose symmetric part is an *hc*-cograph.

The practical usefulness of the directed Fitch relation and its trees depends on how easily the Fitch relation can be estimated from data. Although no convenient tools are available to identify directed xenology relationships without first reconstructing gene and species trees, this does not seem to be a hopeless task at all. The reason is that genes that are imported by HGT from an ancestor of species A into an ancestor of species B, are expected to be more closely related than expected from the bulk of the genome [174, 190]. Since inference from real-life data will never be noise free, it is encouraging that the corresponding editing problem is at least FPT, even though it may be NP-complete as so many other computational problems in phylogenetics.
The directed Fitch relation is the subject of ongoing research: Hellmuth and Seemann [94] recently developed another characterization of the Fitch relation and, using this new characterization, provide an alternative proof of the main Theorem 8.2 of Chapter 8. Moreover, generalizations of Fitch graphs are the subject of [92].

In contrast, the practical implication of the results about the undirected Fitch relation in the context of phylogenetic combinatorics is that the mutual xenology relation cannot convey any interesting phylogenetic information: The only insight that can be gained by considering mutual xenology, is the identification of the maximal subsets of taxa that have not experienced any horizontal transfer events among them. This is due to the fact that the undirected Fitch graphs are exactly the complete multipartite graphs, which in turn are completely defined by their independent sets.

Furthermore, first simulation results (to appear: [211]) evaluating how well the quartet-based workflow presented in Chapter 7 estimates best matches from data, show that the method in its current implementation performs surprisingly poorly. This is most likely due to the choice of the set of outgroups. Practical developments in the near future will thus focus on how this set must be chosen in order to get improved best match estimates. Moreover, ongoing research addresses the question on how to prune the candidate set and on extracting a small set of outgroups to make the procedures fast enough for applications also to large data sets. Besides that, it will also be of interest to investigate how quartet structures and best matches can be used to root a gene tree in case the topology of the corresponding species tree is not known.

The results of this work form the basis for the development of new inference methods relying on (reciprocal) best matches that are suitable for integration into tools such as `Proteinortho`. The theoretical insights into the relationships of (reciprocal) best match graphs, orthology relations, and the estimation of best matches from data as well as insights into the Fitch relation promise drastic improvements in both, the accuracy and the computational performance of RBH-based orthology detection methods.

[1] C. Afrasiabi, B. Samad, D. Dineen, C. Meacham, and K. Sjölander. The PhyloFacts FAT-CAT web server: Ortholog identification and function prediction using fast approximate tree classification. *Nucleic Acids Research*, 41(W1):W242–W248, May 2013. ISSN 0305-1048. doi: 10.1093/nar/gkt399.

[2] A. Aho, M. Garey, and J. Ullman. The Transitive Reduction of a Directed Graph. *SIAM Journal on Computing*, 1(2):131–137, June 1972. ISSN 0097-5397. doi: 10.1137/0201008.

[3] 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(3):405–421, 1981.

[4] Ö. Åkerborg, B. Sennblad, L. Arvestad, and J. Lagergren. Simultaneous Bayesian gene tree reconstruction and reconciliation analysis. *Proceedings of the National Academy of Sciences*, 106(14):5714–5719, 2009. ISSN 0027-8424. doi: 10.1073/pnas.0806251106.

[5] R. Albalat and C. Cañestro. Evolution by gene loss. *Nature Reviews Genetics*, 17(7): 379–391, July 2016. ISSN 1471-0064. doi: 10.1038/nrg.2016.39.

[6] A. M. Altenhoff and C. Dessimoz. Phylogenetic and Functional Assessment of Orthologs Inference Projects and Methods. *PLOS Computational Biology*, 5(1):1–11, Jan. 2009. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000262.

[7] A. M. Altenhoff and C. Dessimoz. Inferring Orthology and Paralogy. In M. Anisimova, editor, *Evolutionary Genomics: Statistical and Computational Methods, Volume 1*, pages 259–279. Humana Press, Totowa, NJ, 2012. ISBN 978-1-61779-582-4.

[8] 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):1–10, May 2012. doi: 10.1371/journal.pcbi.1002514.

[9] A. M. Altenhoff, M. Gil, G. H. Gonnet, and C. Dessimoz. Inferring Hierarchical Orthologous Groups from Orthologous Gene Pairs. *PLOS ONE*, 8(1):1–11, Jan. 2013. doi: 10.1371/journal.pone.0053786.

[10] 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. S. da Silva, D. Szklarczyk, C.-M. Train, P. Bork, O. Lecompte, C. von Mering, I. Xenarios, K. Sjölander, L. J. Jensen, M. J. Martin, M. Muffato, Quest for Orthologs Consortium, A. M. Altenhoff, B. Boeckmann, S. Capella-Gutierrez, T. DeLuca, K. Forslund, J. Huerta-Cepas, B. Linard, C. Pereira, L. P. Pryszcz, F. Schreiber, A. S. da Silva, D. Szklarczyk, C.-M. Train, O. Lecompte, I. Xenarios, K. Sjölander, M. J. Martin, M. Muffato, T. Gabaldón, S. E. Lewis, P. D. Thomas, E. Sonnhammer, C. Dessimoz, T. Gabaldón, S. E. Lewis, P. D. Thomas, E. Sonnhammer, and C. Dessimoz. Standardized benchmarking in the quest for orthologs. *Nature Methods*, 13(5):425–430, May 2016. ISSN 1548-7105. doi: 10.1038/nmeth.3830.

[11] R. I. Aminov and R. I. Mackie. Evolution and ecology of antibiotic resistance genes. *FEMS Microbiology Letters*, 271(2):147–161, June 2007. ISSN 0378-1097. doi: 10.1111/j.1574-6968.2007.00757.x.

[12] J. Anders. *Phylogenetic Trees with Xenology - A New Graph Framework to Represent Horizontal Gene Transfer Events*. Master's Thesis, Leipzig University, Germany, 2017.

[13] L. Arvestad, A.-C. Berglund, J. Lagergren, and B. Sennblad. Bayesian gene/species tree reconciliation and orthology analysis using MCMC. *Bioinformatics*, 19:i7–i15, July 2003. ISSN 1367-4803. doi: 10.1093/bioinformatics/btg1000.

[14] L. Arvestad, J. Lagergren, and B. Sennblad. The Gene Evolution Model and Computing Its Associated Probabilities. *J. ACM*, 56(2):7:1–7:44, Apr. 2009. ISSN 0004-5411. doi: 10.1145/1502793.1502796.

[15] E. Avni, R. Cohen, and S. Snir. Weighted Quartets Phylogenetics. *Systematic Biology*, 64(2):233–242, Mar. 2015. ISSN 1063-5157. doi: 10.1093/sysbio/syu087.

[16] R. K. Azad and J. G. Lawrence. Detecting Laterally Transferred Genes. In M. Anisimova, editor, *Evolutionary Genomics: Statistical and Computational Methods, Volume 1*, Methods in Molecular Biology, pages 281–308. Humana Press, Totowa, NJ, 2012. ISBN 978-1-61779-582-4. doi: 10.1007/978-1-61779-582-4_10.

[17] M. S. Bansal and O. Eulenstein. Algorithms for Genome-Scale Phylogenetics Using Gene Tree Parsimony. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(4):939–956, July 2013. ISSN 1545-5963. doi: 10.1109/TCBB.2013.103.

[18] M. S. Bansal, E. J. Alm, and M. Kellis. Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. *Bioinformatics*, 28(12): i283–i291, June 2012. ISSN 1367-4803. doi: 10.1093/bioinformatics/bts225.

[19] A.-C. Berglund, E. Sjölund, G. Östlund, and E. L. L. Sonnhammer. InParanoid 6: Eukaryotic ortholog clusters with inparalogs. *Nucleic Acids Research*, 36:D263–D266, Nov. 2007. ISSN 0305-1048. doi: 10.1093/nar/gkm1020.

[20] J. Bergsten. A review of long-branch attraction. *Cladistics*, 21(2):163–193, 2005. ISSN 1096-0031. doi: 10.1111/j.1096-0031.2005.00059.x.

[21] S. Böcker and A. W. M. Dress. Recovering Symbolically Dated, Rooted Trees from Symbolic Ultrametrics. *Advances in Mathematics*, 138(1):105–125, Sept. 1998. ISSN 0001-8708. doi: 10.1006/aima.1998.1743.

[22] S. Böcker, S. Briesemeister, and G. W. Klau. Exact algorithms for cluster editing: Evaluation and experiments. *Algorithmica*, 60:316–334, 2011. doi: 10.1007/s00453-009-9339-7.

[23] P. Bork, T. Dandekar, Y. Diaz-Lazcoz, F. Eisenhaber, M. Huynen, and Y. Yuan. Predicting function: From genes to genomes and back. *Journal of Molecular Biology*, 283 (4):707–725, 1998. ISSN 0022-2836. doi: https://doi.org/10.1006/jmbi.1998.2144.

[24] A. Bretscher, D. Corneil, M. Habib, and C. Paul. A Simple Linear Time LexBFS Cograph Recognition Algorithm. *SIAM Journal on Discrete Mathematics*, 22(4):1277–1296, Jan. 2008. ISSN 0895-4801. doi: 10.1137/060664690.

[25] D. Bryant. *Building Trees, Hunting for Trees, and Comparing Trees : Theory and Methods in Phylogenetic Analysis.* PhD thesis, University of Canterbury, 1997.

[26] D. Bryant and M. Steel. Extension Operations on Sets of Leaf-Labeled Trees. *Advances in Applied Mathematics*, 16(4):425–453, Dec. 1995. ISSN 01968858. doi: 10.1006/aama.1995.1020.

[27] J. J. Bull and C. M. Pease. Combinatorics and Variety of Mating-Type Systems. *Evolution*, 43(3):667–671, 1989. ISSN 0014-3820. doi: 10.2307/2409070.

[28] P. Buneman. Note on the Metric Properties of Trees. *J. Comb. Th. B*, 17:48–50, 1974. doi: 10.1016/0095-8956(74)90047-1.

[29] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, May 1996. ISSN 0020-0190. doi: 10.1016/0020-0190(96)00050-6.

[30] W.-C. Chang, P. Górecki, and O. Eulenstein. Exact Solutions for Species Tree Inference from discordant Gene Trees. *Journal of Bioinformatics and Computational Biology*, 11 (05):1342005, 2013. doi: 10.1142/S0219720013420055.

[31] M. Charleston. Jungles: A new solution to the host/parasite phylogeny reconciliation problem. *Mathematical Biosciences*, 149(2):191–223, 1998. ISSN 0025-5564. doi: https://doi.org/10.1016/S0025-5564(97)10012-8.

[32] C. Chauve and N. El-Mabrouk. New Perspectives on Gene Family Evolution: Losses in Reconciliation and a Link with Supertrees. In S. Batzoglou, editor, *Research in Computational Molecular Biology*, pages 46–58. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-02008-7.

[33] F. Chen, A. J. Mackey, J. Stoeckert, Christian J., and D. S. Roos. OrthoMCL-DB: Querying a comprehensive multi-species collection of ortholog groups. *Nucleic Acids Research*, 34:D363–D368, Jan. 2006. ISSN 0305-1048. doi: 10.1093/nar/gkj123.

[34] S. Cherlin, T. M. W. Nye, R. J. Boys, S. E. Heaps, T. A. Williams, and T. M. Embley. The effect of non-reversibility on inferring rooted phylogenies. *Mol Biol Evol*, 35:984–1002, 2018. doi: 10.1093/molbev/msx294.

[35] B. Cloteaux, M. D. LaMar, E. Moseman, and J. Shook. Threshold Digraphs. *J. Res. Natl. Inst. Standards Technology*, 119:227–234, 2014. doi: 10.6028/jres.119.007.

[36] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, July 2009. ISBN 978-0-262-53305-8.

[37] D. Corneil, Y. Perl, and L. Stewart. A Linear Recognition Algorithm for Cographs. *SIAM Journal on Computing*, 14(4):926–934, Nov. 1985. ISSN 0097-5397. doi: 10.1137/0214065.

[38] D. G. Corneil, H. Lerchs, and L. Steward Burlingham. Complement reducible graphs. *Discr. Appl. Math.*, 3:163–174, 1981. doi: 10.1016/0166-218X(81)90013-5.

[39] N. R. Council. *A New Biology for the 21st Century*. The National Academies Press, Washington, DC, 2009. ISBN 978-0-309-14488-9. doi: 10.17226/12764.

[40] C. Crespelle and C. Paul. Fully dynamic recognition algorithm and certificate for directed cographs. *Discrete Applied Mathematics*, 154(12):1722–1741, July 2006. ISSN 0166-218X. doi: 10.1016/j.dam.2006.03.005.

[41] D. A. Dalquen and C. Dessimoz. Bidirectional best hits miss many orthologs in duplication-rich clades such as plants and animals. *Genome Biol Evol*, 5:1800–1806, 2013. doi: 10.1093/gbe/evt132.

[42] D. A. Dalquén, M. Anisimova, G. H. Gonnet, and C. Dessimoz. ALF – A Simulation Framework for Genome Evolution. *Mol. Biol. Evol.*, 29:1115–1123, 2011. doi: 10.1093/molbev/msr268.

[43] C. Darwin. *On the Origin of Species by Means of Natural Selection, or the Preservation of Favored Races in the Struggle for Life*. J. Murray, London, 1859.

[44] V. Daubin, E. Lerat, and G. Perrière. The source of laterally transferred genes in bacterial genomes. *Genome Biology*, 4(9):R57, Aug. 2003. ISSN 1474-760X. doi: 10.1186/gb-2003-4-9-r57.

[45] M. C. H. Dekker. *Reconstruction Methods for Derivation Trees*. Master's Thesis, Vrije Universiteit, Amsterdam, Netherlands, 1986.

[46] T. F. DeLuca, I.-H. Wu, J. Pu, S. Singh, T. Monaghan, L. Peshkin, and D. P. Wall. Roundup: A multi-genome repository of orthologs and evolutionary distances. *Bioinformatics*, 22(16):2044–2046, June 2006. ISSN 1367-4803. doi: 10.1093/bioinformatics/btl286.

[47] Y. Deng and D. Fernández-Baca. Fast Compatibility Testing for Rooted Phylogenetic Trees. *Algorithmica*, 80(8):2453–2477, Aug. 2018. ISSN 1432-0541. doi: 10.1007/s00453-017-0330-4.

[48] 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 *Comparative Genomics*, Lecture Notes in Computer Science, pages 61–72. Springer, Berlin, Heidelberg, Sept. 2005. ISBN 978-3-540-28932-6 978-3-540-31814-9. doi: 10.1007/11554714_6.

[49] 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(11):3309–3316, Jan. 2006. ISSN 0305-1048. doi: 10.1093/nar/gkl433.

[50] C. Dessimoz, D. Margadant, and G. H. Gonnet. DLIGHT – Lateral Gene Transfer Detection Using Pairwise Evolutionary Distances in a Statistical Framework. In M. Vingron and L. Wong, editors, *Research in Computational Molecular Biology*, pages 315–330. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-78839-3.

[51] T. Dobzhansky. Nothing in Biology Makes Sense Except in the Light of Evolution. *The American Biology Teacher*, 75(2):87–91, Feb. 2013. ISSN 0002-7685, 1938-4211. doi: 10.2307/4444260.

[52] R. Dondi, N. El-Mabrouk, and M. Lafond. Correction of Weighted Orthology and Paralogy Relations-Complexity and Algorithmic Results. In *International Workshop on Algorithms in Bioinformatics*, pages 121–136. Springer, Springer International Publishing, 2016. ISBN 978-3-319-43681-4.

[53] R. Dondi, M. Lafond, and N. El-Mabrouk. Approximating the correction of weighted and unweighted orthology and paralogy relations. *Algorithms for Molecular Biology*, 12 (1):4, 2017. doi: 10.1186/s13015-017-0096-x.

[54] R. Dondi, G. Mauri, and I. Zoppis. Orthology Correction for Gene Tree Reconstruction: Theoretical and Experimental Results. *Procedia Computer Science*, 108:1115–1124, 2017. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2017.05.047.

[55] 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, Oct. 2009. ISSN 1066-5277, 1557-8666. doi: 10.1089/cmb.2009.0095.

[56] J.-P. Doyon, C. Scornavacca, K. Gorbunov, G. Szöllősi, V. Ranwez, and V. Berry. An Efficient Algorithm for Gene/Species Trees Parsimonious Reconciliation with Losses, Duplications and Transfers. In *Comparative Genomics: International Workshop, RECOMB-CG 2010, Ottawa, Canada, October 9-11, 2010. Proceedings*, pages 93–108. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[57] J.-P. Doyon, V. Ranwez, V. Daubin, and V. Berry. Models, algorithms and programs for phylogeny reconciliation. *Briefings in Bioinformatics*, 12(5):392–400, Sept. 2011. ISSN 1467-5463. doi: 10.1093/bib/bbr045.

[58] A. J. Drummond, S. Y. W. Ho, M. J. Phillips, and A. Rambaut. Relaxed phylogenetics and dating with confidence. *PLoS Biol*, 4:699–710, 2006. doi: 10.1371/journal.pbio.0040088.

[59] I. Ebersberger, S. Strauss, and A. von Haeseler. HaMStR: Profile hidden markov model based search for orthologs in ESTs. *BMC Evolutionary Biology*, 9(1):157, July 2009. ISSN 1471-2148. doi: 10.1186/1471-2148-9-157.

[60] A. Ehrenfeucht and G. Rozenberg. Primitivity is hereditary for 2-structures. *Theoretical Computer Science*, 70(3):343–358, Feb. 1990. ISSN 0304-3975. doi: 10.1016/0304-3975(90)90131-Z.

[61] M. Eigen, R. Winkler-Oswatitsch, and A. W. M. Dress. Statistical geometry in sequence space: A method of quantitative comparative sequence analysis. *Proc Natl Acad Sci USA*, 85:5913–5917, 1988.

[62] A. Elmasry. The Subset Partial Order: Computing and Combinatorics. In *2010 Proceedings of the Seventh Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, Proceedings, pages 27–33. Society for Industrial and Applied Mathematics, Jan. 2010. ISBN 978-0-89871-933-8. doi: 10.1137/1.9781611973006.4.

[63] D. M. Emms and S. Kelly. OrthoFinder: Solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. *Genome Biology*, 16 (1):157, Aug. 2015. ISSN 1474-760X. doi: 10.1186/s13059-015-0721-2.

[64] W. M. Fitch. Distinguishing Homologous from Analogous Proteins. *Systematic Biology*, 19(2):99–113, June 1970. ISSN 1063-5157. doi: 10.2307/2412448.

[65] W. M. Fitch. A non-sequential method for constructing trees and hierarchical classifications. *Journal of Molecular Evolution*, 18(1):30–37, Jan. 1981. ISSN 1432-1432. doi: 10.1007/BF01733209.

[66] W. M. Fitch. Homology: A personal view on some of the problems. *Trends in Genetics*, 16(5):227–231, 2000. ISSN 0168-9525. doi: 10.1016/S0168-9525(00)02005-9.

[67] W. T. Fitch. Glossogeny and phylogeny: Cultural evolution meets genetic evolution. *Trends in Genetics*, 24(8):373–374, 2008. ISSN 0168-9525. doi: https://doi.org/10.1016/j.tig.2008.05.003.

[68] 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: 1531–1545, 1999.

[69] D. L. Fulton, Y. Y. Li, M. R. Laird, B. G. Horsman, F. M. Roche, and F. S. Brinkman. Improving the specificity of high-throughput ortholog prediction. *BMC Bioinformatics*, 7(1):270, May 2006. ISSN 1471-2105. doi: 10.1186/1471-2105-7-270.

[70] H. N. Gabow and R. E. Tarjan. A linear-time algorithm for a special case of disjoint set union. In *Proceedings of the 15th ACM Symposium on Theory of Computing (STOC)*, pages 246–251. ACM, 1983. doi: 10.1145/800061.808753.

[71] Y. Gao, D. R. Hare, and J. Nastos. The cluster deletion problem for cographs. *Discrete Mathematics*, 313(23):2763–2771, 2013. ISSN 0012-365X. doi: https://doi.org/10.1016/j.disc.2013.08.017.

[72] 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(5):1459–1491, June 2018. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-018-1260-8.

[73] 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(7):2015–2057, Apr. 2019. ISSN 1432-1416. doi: 10.1007/s00285-019-01332-9.

[74] M. Geiß, M. González, A. López, D. I. Valdivia, M. Hellmuth, M. Hernández Rosales, and P. F. Stadler. Best Match Graphs and Reconciliation of Gene Trees with Species Trees. *arXiv:1904.12021*, Apr. 2019.

[75] M. Geiß, M. Hellmuth, and P. F. Stadler. Reciprocal Best Match Graphs. *arXiv:1903.07920*, Mar. 2019.

[76] J. A. Gerlt and P. C. Babbitt. Can sequence determine function? *Genome Biology*, 1 (5):reviews0005.1, Nov. 2000. ISSN 1474-760X. doi: 10.1186/gb-2000-1-5-reviews0005.

[77] 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, June 1979. ISSN 1063-5157. doi: 10.1093/sysbio/28.2.132.

[78] P. Górecki and O. Eulenstein. Algorithms: Simultaneous error-correction and rooting for gene tree reconciliation and the gene duplication problem. *BMC Bioinformatics*, 13 (10):S14, June 2012. doi: 10.1186/1471-2105-13-S10-S14.

[79] P. Górecki and J. Tiuryn. DLS-trees: A model of evolutionary scenarios. *Theoretical Computer Science*, 359(1):378–399, Aug. 2006. ISSN 0304-3975. doi: 10.1016/j.tcs.2006.05.019.

[80] P. Górecki, G. J. Burleigh, and O. Eulenstein. Maximum likelihood models and algorithms for gene tree evolution with duplications and losses. *BMC Bioinformatics*, 12 (1):S15, Feb. 2011. ISSN 1471-2105. doi: 10.1186/1471-2105-12-S1-S15.

[81] P. Górecki, O. Eulenstein, and J. Tiuryn. Unrooted Tree Reconciliation: A Unified Approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10 (2):522–536, Mar. 2013. ISSN 1545-5963. doi: 10.1109/TCBB.2013.22.

[82] P. Górecki, A. Mykowiecka, J. Paszek, and O. Eulenstein. Mathematical properties of the gene duplication cost. *Discrete Applied Mathematics*, Dec. 2018. ISSN 0166-218X. doi: 10.1016/j.dam.2018.11.014.

[83] D. Gries, A. J. Martin, J. L. A. van de Snepscheut, and J. T. Udding. An algorithm for transitive reduction of an acyclic graph. *Sci. Computer Prog.*, 12:151–155, 1989. doi: 10.1016/0167-6423(89)90039-7.

[84] S. Grünewald, M. Steel, and M. S. Swenson. Closure operations in phylogenetics. *Mathematical Biosciences*, 208(2):521–537, Aug. 2007. ISSN 0025-5564. doi: 10.1016/j.mbs.2006.11.005.

[85] R. Guigo, I. Muchnik, and T. F. Smith. Reconstruction of Ancient Molecular Phylogeny. *Molecular Phylogenetics and Evolution*, 6(2):189–213, Oct. 1996. ISSN 1055-7903. doi: 10.1006/mpev.1996.0071.

[86] M. Habib and C. Paul. A simple linear time algorithm for cograph recognition. *Discrete Applied Mathematics*, 145(2):183–197, Jan. 2005. ISSN 0166-218X. doi: 10.1016/j.dam.2004.01.011.

[87] R. Hammack, W. Imrich, S. Klavžar, W. Imrich, and S. Klavžar. *Handbook of Product Graphs*. CRC Press, June 2011. ISBN 978-0-429-13059-5. doi: 10.1201/b10959.

[88] F. Harary and A. J. Schwenk. The number of caterpillars. *Discrete Mathematics*, 6(4): 359–365, Jan. 1973. ISSN 0012-365X. doi: 10.1016/0012-365X(73)90067-8.

[89] D. Harel and R. Tarjan. Fast Algorithms for Finding Nearest Common Ancestors. *SIAM Journal on Computing*, 13(2):338–355, May 1984. ISSN 0097-5397. doi: 10.1137/0213024.

[90] M. Hasegawa, H. Kishino, and T. Yano. Dating of human-ape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.*, 22:160–174, 1985. doi: 10.1007/BF02101694.

[91] M. Hellmuth. Biologically feasible gene trees, reconciliation maps and informative triples. *Algorithms for Molecular Biology*, 12(1):23, Aug. 2017. ISSN 1748-7188. doi: 10.1186/s13015-017-0114-z.

[92] M. Hellmuth. Generalized Fitch graphs: Edge-labeled graphs that are explained by edge-labeled trees. *Discrete Applied Mathematics*, June 2019. ISSN 0166-218X. doi: 10.1016/j.dam.2019.06.015.

[93] M. Hellmuth and T. Marc. On the Cartesian skeleton and the factorization of the strong product of digraphs. *Theor Comp Sci*, 565:16–29, 2015. doi: 10.1016/j.tcs.2014.10.045.

[94] M. Hellmuth and C. R. Seemann. Alternative characterizations of Fitch's xenology relation. *Journal of Mathematical Biology*, May 2019. ISSN 1432-1416. doi: 10.1007/s00285-019-01384-x.

[95] M. Hellmuth and N. Wieseke. On Symbolic Ultrametrics, Cotree Representations, and Cograph Edge Decompositions and Partitions. In *Computing and Combinatorics*, Lecture Notes in Computer Science, pages 609–623. Springer, Cham, Aug. 2015. ISBN 978-3-319-21398-9. doi: 10.1007/978-3-319-21398-9_48.

[96] M. Hellmuth and N. Wieseke. From Sequence Data incl. Orthologs, Paralogs, and Xenologs to Gene and Species Trees. In *Evolutionary Biology*, pages 373–392. Springer International Publishing, 2016.

[97] 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(1-2):399–420, Jan. 2013. ISSN 0303-6812, 1432-1416. doi: 10.1007/s00285-012-0525-x.

[98] M. Hellmuth, N. Wieseke, M. Lechner, H.-P. Lenhof, M. Middendorf, and P. F. Stadler. Phylogenetics from Paralogs. *Proceedings of the National Academy of Sciences*, 112(7):2058–2063, Feb. 2015. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1412770112.

[99] M. Hellmuth, A. Fritz, N. Wieseke, and P. F. Stadler. Cograph Editing: Merging Modules is equivalent to Editing P4's. *arXiv preprint arXiv:1702.07499*, 2017.

[100] 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(1):199–237, July 2017. ISSN 1432-1416. doi: 10.1007/s00285-016-1084-3.

[101] 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(1):#P1.08, 2018. doi: 10.26493/2590-9770.1245.98c.

[102] M. Hellmuth, M. Geiß, and P. F. Stadler. Complexity of Modification Problems for Reciprocal Best Match Graphs. *arXiv:1907.08865*, July 2019.

[103] M. Hellmuth, K. Huber, and V. Moulton. Reconciling Event-Labeled Gene Trees with MUL-trees and Species Networks. *J. Math. Biology*, 2019.

[104] M. R. Henzinger, V. King, and T. Warnow. Constructing a Tree from Homeomorphic Subtrees, with Applications to Computational Evolutionary Biology. *Algorithmica*, 24 (1):1–13, May 1999. ISSN 1432-0541. doi: 10.1007/PL00009268.

[105] 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(19):S6, Dec. 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-S19-S6.

[106] P. N. Hess and C. A. de Moraes Russo. An empirical test of the midpoint rooting method. *Biol. J. Linnean Soc.*, 92:669–674, 2007. doi: 10.1111/j.1095-8312.2007.00864.x.

[107] M. Hiller, B. T. Schaar, V. B. Indjeian, D. M. Kingsley, L. R. Hagey, and G. Bejerano. A "Forward Genomics" Approach Links Genotype to Phenotype using Independent Phenotypic Losses among Related Species. *Cell Reports*, 2(4):817–823, 2012. ISSN 2211-1247. doi: https://doi.org/10.1016/j.celrep.2012.08.032.

[108] C. T. Hoàngm, M. Kamiński, J. Sawada, and R. Sritharan. Finding and listing induced paths and cycles. *Discr. Appl. Math.*, 161:633–641, 2013. doi: 10.1016/j.dam.2012.01.024.

[109] B. R. Holland, D. Penny, and M. D. Hendy. Outgroup misplacement and phylogenetic inaccuracy under a molecular clock — a simulation study. *Syst. Biol*, 52:229–238, 2003. doi: 10.1080/10635150390192771.

[110] J. Holm and K. D. Lichtenberg. Poly-Logarithmic Deterministic Fully-Dynamic Algorithms for Connectivity, Minimum Spanning Tree, 2-Edge, and Biconnectivity. *Journal of the ACM (JACM)*, 48(4):723–760, 2001.

[111] 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, Sept. 2002. ISSN 1063-5157. doi: 10.1080/10635150290102366.

[112] J. Huerta-Cepas, L. P. Pryszcz, and T. Gabaldón. MetaPhOrs: Orthology and paralogy predictions from multiple phylogenetic evidence using a consistency-based confidence score. *Nucleic Acids Research*, 39(5):e32–e32, Dec. 2010. ISSN 0305-1048. doi: 10.1093/nar/gkq953.

[113] J. Huerta-Cepas, L. P. Pryszcz, M. Marcet-Houben, S. Capella-Gutiérrez, and T. Gabaldón. PhylomeDB v4: Zooming into the plurality of evolutionary histories of a genome. *Nucleic Acids Research*, 42(D1):D897–D902, Nov. 2013. ISSN 0305-1048. doi: 10.1093/nar/gkt1177.

[114] D. H. Huson, R. Rupp, and C. Scornavacca. *Phylogenetic Networks: Concepts, Algorithms and Applications*. Cambridge University Press, Dec. 2010. ISBN 978-1-139-49287-4.

[115] H. Innan and F. Kondrashov. The evolution of gene duplications: Classifying and distinguishing between models. *Nature Reviews Genetics*, 11(2):97, 2010.

[116] S. Jahangiri-Tazehkand, L. Wong, and C. Eslahchi. OrthoGNC: A Software for Accurate Identification of Orthologs Based on Gene Neighborhood Conservation. *Genomics Proteomics Bioinformatics*, 15:361–370, 2017. doi: 10.1016/j.gpb.2017.07.002.

[117] B. Jamison and S. Olariu. Recognizing P_4-sparse graphs in linear time. *SIAM J. Computing*, 21:381–406, 1992. doi: 10.1137/0221027.

[118] J. Jansson, J. H.-K. Ng, K. Sadakane, and W.-K. Sung. Rooted Maximum Agreement Supertrees. *Algorithmica*, 43(4):293–307, Dec. 2005. ISSN 1432-0541. doi: 10.1007/s00453-004-1147-5.

[119] L. J. Jensen, P. Julien, M. Kuhn, C. von Mering, J. Muller, T. Doerks, and P. Bork. eggNOG: Automated construction and annotation of orthologous groups of genes. *Nucleic Acids Research*, 36:D250–D254, Oct. 2007. ISSN 0305-1048. doi: 10.1093/nar/gkm796.

[120] R. A. Jensen. Orthologs and paralogs - we need to get it right. *Genome Biology*, 2(8):interactions1002, Aug. 2001. ISSN 1474-760X. doi: 10.1186/gb-2001-2-8-interactions1002.

[121] A. L. Juárez-Vázquez, J. N. Edirisinghe, E. A. Verduzco-Castro, K. Michalska, C. Wu, L. Noda-García, G. Babnigg, M. Endres, S. Medina-Ruíz, J. Santoyo-Flores, M. Carrillo-Tripp, H. Ton-That, A. Joachimiak, C. S. Henry, and F. Barona-Gómez. Evolution of substrate specificity in a retained enzyme driven by gene loss. *eLife*, 6, Mar. 2017. ISSN 2050-084X. doi: 10.7554/eLife.22679.

[122] T. H. Jukes and C. R. Cantor. Evolution of protein molecules. In H. N. Munro, editor, *Mammalian Protein Metabolism*, pages 21–132. Academic Press, New York, 1969.

[123] M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Research*, 44(D1):D457–D462, Oct. 2015. ISSN 0305-1048. doi: 10.1093/nar/gkv1070.

[124] L. A. Katz, J. R. Grant, L. W. Parfrey, and J. G. Burleigh. Turning the crown upside down: Gene tree parsimony roots the eukaryotic tree of life. *Syst. Biol.*, 61:653–660, 2012. doi: 10.1093/sysbio/sys026.

[125] P. J. Keeling and J. D. Palmer. Horizontal gene transfer in eukaryotic evolution. *Nature Reviews Genetics*, 9(8):605–618, Aug. 2008. ISSN 1471-0064. doi: 10.1038/nrg2386.

[126] S. Keller-Schmidt and K. Klemm. A Model of Macroevolution As a Branching Process Based on Innovations. *Adv. Complex Syst.*, 15:1250043, 2012. doi: 10.1142/S0219525912500439.

[127] M. Kellis, N. Patterson, M. Endrizzi, B. Birren, and E. S. Lander. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, 423 (6937):241, May 2003. ISSN 1476-4687. doi: 10.1038/nature01644.

[128] M. Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*, 16:111–120, 1980. doi: 10.1007/BF01731581.

[129] T. Kinene, J. Wainaina, S. Maina, and L. Boykin. Rooting Trees, Methods for. In R. M. Kliman, editor, *Encyclopedia of Evolutionary Biology*, volume 3, page 489. Elsevier, Amsterdam, NL, 2016. doi: 10.1016/B978-0-12-800049-6.00215-8.

[130] E. V. Koonin. An apology for orthologs - or brave new memes. *Genome Biology*, 2(4): comment1005.1, Apr. 2001. ISSN 1474-760X. doi: 10.1186/gb-2001-2-4-comment1005.

[131] E. V. Koonin. Orthologs, Paralogs, and Evolutionary Genomics. *Annual Review of Genetics*, 39(1):309–338, 2005. doi: 10.1146/annurev.genet.39.073003.114725.

[132] E. V. Koonin, K. S. Makarova, and L. Aravind. Horizontal Gene Transfer in Prokaryotes: Quantification and Classification. *Annual Review of Microbiology*, 55(1):709–742, 2001. doi: 10.1146/annurev.micro.55.1.709.

[133] N. Krishnamurthy, D. P. Brown, D. Kirshner, and K. Sjölander. PhyloFacts: An online structural phylogenomic encyclopedia for protein functional and structural classification. *Genome Biology*, 7(9):R83, Sept. 2006. ISSN 1474-760X. doi: 10.1186/gb-2006-7-9-r83.

[134] 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, June 2011. ISSN 1467-5463. doi: 10.1093/bib/bbr030.

[135] 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(D1): D250–D256, Nov. 2014. ISSN 0305-1048. doi: 10.1093/nar/gku1220.

[136] T. S. Kuhn, A. Ø. Mooers, and G. H. Thomas. A simple polytomy resolver for dated phylogenies. *Methods Ecol. Evo.*, 2:427–436, 2011. doi: 10.1111/j.2041-210X.2011.00103.x.

[137] S. Kumar. Molecular clocks: Four decades of evolution. *Nat Rev Genet*, 6:654–662, 2005. doi: 10.1038/nrg1659.PMID16136655.

[138] M. Lafond and N. El-Mabrouk. Orthology and paralogy constraints: Satisfiability and consistency. *BMC Genomics*, 15(6):S12, Oct. 2014. ISSN 1471-2164. doi: 10.1186/1471-2164-15-S6-S12.

[139] M. Lafond and N. El-Mabrouk. Orthology Relation and Gene Tree Correction: Complexity Results. In *International Workshop on Algorithms in Bioinformatics*, pages 66–79. Springer Berlin Heidelberg, 2015. ISBN 978-3-662-48221-6.

[140] M. Lafond, K. M. Swenson, and N. El-Mabrouk. An Optimal Reconciliation Algorithm for Gene Trees with Polytomies. In B. Raphael and J. Tang, editors, *Algorithms in Bioinformatics*, pages 106–122. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33122-0.

[141] M. Lafond, R. Dondi, and N. El-Mabrouk. The link between orthology relations and gene trees: A correction perspective. *Algorithms for Molecular Biology*, 11(1):4, Apr. 2016. ISSN 1748-7188. doi: 10.1186/s13015-016-0067-7.

[142] J. G. Lawrence and D. L. Hartl. Inference of horizontal genetic transfer from molecular data: An approach using the bootstrap. *Genetics*, 131(3):753–760, 1992. ISSN 0016-6731.

[143] J. G. Lawrence and H. Ochman. Amelioration of Bacterial Genomes: Rates of Change and Exchange. *Journal of Molecular Evolution*, 44(4):383–397, Apr. 1997. ISSN 1432-1432. doi: 10.1007/PL00006158.

[144] 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 (1):124, Apr. 2011. ISSN 1471-2105. doi: 10.1186/1471-2105-12-124.

[145] 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(8):1–10, Aug. 2014. ISSN 1932-6203. doi: 10.1371/journal.pone.0105015.

[146] P. Lemey, M. Salemi, and A.-M. Vandamme. *The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing*. Cambridge University Press, Mar. 2009. ISBN 978-1-139-47861-8.

[147] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, Apr. 1980. ISSN 0022-0000. doi: 10.1016/0022-0000(80)90060-4.

[148] J. Li. Combinatorial Logarithm and Point-Determining Cographs. *Elec. J. Comb.*, 19: P8, 2012.

[149] Y. Liu, J. Wang, J. Guo, and J. Chen. Cograph Editing: Complexity and Parameterized Algorithms. In B. Fu and D.-Z. Du, editors, *Computing and Combinatorics*, pages 110–121. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-22685-4.

[150] Y. Liu, J. Wang, J. Guo, and J. Chen. Complexity and parameterized algorithms for Cograph Editing. *Theoretical Computer Science*, 461:45–54, 2012. ISSN 0304-3975. doi: https://doi.org/10.1016/j.tcs.2011.11.040.

[151] A. López Sánchez. *ESTUDIO COMPUTACIONAL DE ESCENARIOS EVOLUTIVOS*. Bachelor's Thesis, UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, Mexico, 2019.

[152] J. B. Losos, S. J. Arnold, G. Bejerano, E. D. B. Iii, D. Hibbett, H. E. Hoekstra, D. P. Mindell, A. Monteiro, C. Moritz, H. A. Orr, D. A. Petrov, S. S. Renner, R. E. Ricklefs, P. S. Soltis, and T. L. Turner. Evolutionary Biology for the 21st Century. *PLOS Biology*, 11(1):e1001466, Jan. 2013. ISSN 1545-7885. doi: 10.1371/journal.pbio.1001466.

[153] B. Ma, M. Li, and L. Zhang. From Gene Trees to Species Trees. *SIAM Journal on Computing*, 30(3):729–752, Jan. 2000. ISSN 0097-5397. doi: 10.1137/S0097539798343362.

[154] W. P. Maddison. Gene Trees in Species Trees. *Systematic Biology*, 46(3):523–536, Sept. 1997. ISSN 1063-5157. doi: 10.1093/sysbio/46.3.523.

[155] U. Mai, E. Sayyari, and S. Mirarab. Minimum variance rooting of phylogenetic trees and implications for species tree reconstruction. *PLOS ONE*, 12(8):e0182238, Aug. 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0182238.

[156] E. M. McCarthy. *On the Origins of New Forms of Life A New Theory*. 2008.

[157] R. M. McConnell and F. de Montgolfier. Linear-time modular decomposition of directed graphs. *Discrete Applied Mathematics*, 145(2):198–209, Jan. 2005. ISSN 0166-218X. doi: 10.1016/j.dam.2004.02.017.

[158] R. M. McConnell and J. P. Spinrad. Modular decomposition and transitive orientation. *Discrete Mathematics*, 201(1):189–241, Apr. 1999. ISSN 0012-365X. doi: 10.1016/S0012-365X(98)00319-7.

[159] R. McKenzie. Cardinal multiplication of structures with a reflexive relation. *Fundamenta Mathematicae*, 70(1):59–101, 1971. ISSN 0016-2736.

[160] C. Médigue, T. Rouxel, P. Vigier, A. Hénaut, and A. Danchin. Evidence for horizontal gene transfer in Escherichia coli speciation. *Journal of Molecular Biology*, 222(4):851–856, Dec. 1991. ISSN 0022-2836. doi: 10.1016/0022-2836(91)90575-Q.

[161] D. Merkle and M. Middendorf. Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information. *Theory in Biosciences*, 123(4):277–299, Apr. 2005. ISSN 1611-7530. doi: 10.1016/j.thbio.2005.01.003.

[162] R. H. Möhring and F. J. Radermacher. Substitution Decomposition for Discrete Structures and Connections with Combinatorial Optimization. In *North-Holland Mathematics Studies*, volume 95 of *Algebraic and Combinatorial Methods in Operations Research*, pages 257–355. North-Holland, Jan. 1984. doi: 10.1016/S0304-0208(08)72966-9.

[163] G. Moreno-Hagelsieb and K. Latimer. Choosing BLAST options for better detection of orthologs as reciprocal best hits. *Bioinformatics*, 24(3):319–324, Nov. 2007. ISSN 1367-4803. doi: 10.1093/bioinformatics/btm585.

[164] I. Moszer, E. P. Rocha, and A. Danchin. Codon usage and lateral gene transfer in Bacillus subtilis. *Current Opinion in Microbiology*, 2(5):524–528, Oct. 1999. ISSN 1369-5274. doi: 10.1016/S1369-5274(99)00011-9.

[165] S. A. Muhammad, B. Sennblad, and J. Lagergren. Species tree-aware simultaneous reconstruction of gene and domain evolution. *bioRxiv*, 2018. doi: 10.1101/336453.

[166] Y. Nakatani, H. Takeda, Y. Kohara, and S. Morishita. Reconstruction of the vertebrate ancestral genome reveals dynamic genome reorganization in early vertebrates. *Genome Research*, 17(9):1254–1265, Jan. 2007. ISSN 1088-9051, 1549-5469. doi: 10.1101/gr.6316407.

[167] 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(6):1–10, June 2011. doi: 10.1371/journal.pcbi.1002073.

[168] M. Nei and S. Kumar. *Molecular Evolution and Phylogenetics*. Oxford university press, 2000.

[169] 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.

[170] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[171] K. Nieselt-Struwe. Graphs in sequence spaces: A review of statistical geometry. *Biophys Chem.*, 66:111–131, 1997.

[172] K. Nieselt-Struwe and A. von Haeseler. Quartet-Mapping, a Generalization of the Likelihood-Mapping Procedure. *Molecular Biology and Evolution*, 18(7):1204–1219, July 2001. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a003907.

[173] 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(1):2, Feb. 2018. ISSN 1748-7188. doi: 10.1186/s13015-018-0121-8.

[174] P. S. Novichkov, M. V. Omelchenko, M. S. Gelfand, 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(19):6575–6585, Oct. 2004. ISSN 0021-9193, 1098-5530. doi: 10.1128/JB.186.19.6575-6585.2004.

[175] T. O'Connor, K. Sundberg, H. Carroll, M. Clement, and Q. Snell. Analysis of long branch extraction and long branch shortening. *BMC Genomics*, 11(2):S14, Nov. 2010. ISSN 1471-2164. doi: 10.1186/1471-2164-11-S2-S14.

[176] S. Ohno. *Evolution by Gene Duplication*. Springer Science & Business Media, Dec. 2013. ISBN 978-3-642-86659-3.

[177] R. Overbeek, M. Fonstein, M. D, G. D. Pusch, and N. Maltsev. The use of gene clusters to infer functional coupling. *Proceedings of the National Academy of Sciences*, 96(6): 2896–2901, 1999. ISSN 0027-8424. doi: 10.1073/pnas.96.6.2896.

[178] R. D. M. Page. Maps Between Trees and Cladistic Analysis of Historical Associations among Genes,Organisms, and Areas. *Systematic Biology*, 43(1):58–77, Mar. 1994. ISSN 1063-5157. doi: 10.1093/sysbio/43.1.58.

[179] R. D. M. Page and J. A. Cotton. Vertebrate phylogenomics: Reconciled trees and gene duplications. *Pac Symp Biocomput*, pages 536–547, 2002. doi: 10.1142/9789812799623_0050.

[180] A. C. Palmer and R. Kishony. Understanding, predicting and manipulating the genotypic evolution of antibiotic resistance. *Nature Reviews Genetics*, 14:243–248, 2013. doi: 10.1038/nrg3351.

[181] J. S. Pedersen, G. Bejerano, A. Siepel, K. Rosenbloom, K. Lindblad-Toh, E. S. Lander, J. Kent, W. Miller, and D. Haussler. Identification and Classification of Conserved RNA Secondary Structures in the Human Genome. *PLOS Computational Biology*, 2(4):e33, Apr. 2006. ISSN 1553-7358. doi: 10.1371/journal.pcbi.0020033.

[182] D. Penny. Criteria for optimising phylogenetic trees and the problem of determining the root of a tree. *Journal of Molecular Evolution*, 8(2):95–116, June 1976. ISSN 1432-1432. doi: 10.1007/BF01739097.

[183] M. Petersen, K. Meusemann, A. Donath, D. Dowling, S. Liu, R. S. Peters, L. Podsiadlowski, A. Vasilikopoulos, X. Zhou, B. Misof, and O. Niehuis. Orthograph: A versatile tool for mapping coding nucleotide sequences to clusters of orthologous genes. *BMC Bioinformatics*, 18(1):111, Feb. 2017. doi: 10.1186/s12859-017-1529-8.

[184] G. A. Petsko. Homologuephobia. *Genome Biology*, 2(2):comment1002.1, Feb. 2001. ISSN 1474-760X. doi: 10.1186/gb-2001-2-2-comment1002.

[185] P. Pritchard. A simple sub-quadratic algorithm for computing the subset partial order. *Information Processing Letters*, 56(6):337–341, Dec. 1995. ISSN 0020-0190. doi: 10.1016/0020-0190(95)00165-4.

[186] S. J. Prohaska and P. F. Stadler. The duplication of the Hox gene clusters in teleost fishes. *Theory in Biosciences*, 123(1):89–110, June 2004. ISSN 1611-7530. doi: 10.1016/j.thbio.2004.03.004.

[187] A. Purvis and T. Garland Jr. Polytomies in comparative analyses of continuous characters. *Syst. Biol.*, 42:569–575, 1993. doi: 10.2307/2992489.

[188] C. Rancurel, L. Legrand, and E. G. J. Danchin. Alienness: Rapid Detection of Candidate Horizontal Gene Transfers across the Tree of Life. *Genes*, 8(10), 2017. ISSN 2073-4425. doi: 10.3390/genes8100248.

[189] M. D. Rasmussen and M. Kellis. A Bayesian Approach for Fast and Accurate Gene Tree Reconstruction. *Molecular Biology and Evolution*, 28(1):273–290, July 2010. ISSN 0737-4038. doi: 10.1093/molbev/msq189.

[190] M. Ravenhall, N. Škunca, F. Lassalle, and C. Dessimoz. Inferring Horizontal Gene Transfer. *PLOS Computational Biology*, 11(5):1–16, May 2015. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004095.

[191] M. Remm, C. E. Storm, and E. L. Sonnhammer. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *Journal of Molecular Biology*, 314(5): 1041–1052, 2001. ISSN 0022-2836. doi: https://doi.org/10.1006/jmbi.2000.5197.

[192] N. Retzlaff and P. F. Stadler. Phylogenetics beyond biology. *Theory in Biosciences*, 137 (2):133–143, Nov. 2018. ISSN 1611-7530. doi: 10.1007/s12064-018-0264-7.

[193] A. C. Roth, G. H. Gonnet, and C. Dessimoz. Algorithm of OMA for large-scale orthology inference. *BMC Bioinformatics*, 9(1):518, Dec. 2008. ISSN 1471-2105. doi: 10.1186/1471-2105-9-518.

[194] L. Y. Rusin, E. V. Lyubetskaya, K. Y. Gorbunov, and V. A. Lyubetsky. Reconciliation of Gene and Species Trees, 2014.

[195] W. Salzburger and A. Meyer. The species flocks of East African cichlid fishes: Recent advances in molecular phylogenetics and population genetics. *Naturwissenschaften*, 91 (6):277–290, June 2004. ISSN 1432-1904. doi: 10.1007/s00114-004-0528-6.

[196] S. Sattath and A. Tversky. Additive similarity trees. *Psychometrika*, 42(3):319–345, Sept. 1977. ISSN 1860-0980. doi: 10.1007/BF02293654.

[197] E. Sayyari and S. Mirarab. Testing for Polytomies in Phylogenetic Species Trees Using Quartet Frequencies. *Genes*, 9:E132, 2018. doi: 10.3390/genes9030132.

[198] B. Schieber and U. Vishkin. On Finding Lowest Common Ancestors: Simplification and Parallelization. *SIAM Journal on Computing*, 17(6):1253–1262, Dec. 1988. ISSN 0097-5397. doi: 10.1137/0217079.

[199] A. Schneider, C. Dessimoz, and G. H. Gonnet. OMA Browser—Exploring orthologous relations across 352 complete genomes. *Bioinformatics*, 23(16):2180–2182, Aug. 2007. ISSN 1367-4803. doi: 10.1093/bioinformatics/btm295.

[200] C. R. Seemann and M. Hellmuth. The matroid structure of representative triple sets and triple-closure computation. *European Journal of Combinatorics*, 70:384–407, May 2018. ISSN 0195-6698. doi: 10.1016/j.ejc.2018.02.013.

[201] C. Semple. Reconstructing minimal rooted trees. *Discrete Applied Mathematics*, 127 (3):489–503, May 2003. ISSN 0166-218X. doi: 10.1016/S0166-218X(02)00250-0.

[202] C. Semple and M. Steel. *Phylogenetics*, volume 24 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, Feb. 2003. ISBN 978-0-19-850942-4.

[203] M. Seo, H. Jeong, H. Kim, K. Caetano-Anollés, S. Sung, S. Cho, T. Kwon, S. H. Choi, and A. Nasir. HGTree: Database of horizontally transferred genes determined by tree reconciliation. *Nucleic Acids Research*, 44(D1):D610–D619, Nov. 2015. ISSN 0305-1048. doi: 10.1093/nar/gkv1245.

[204] J. C. Setubal and P. F. Stadler. Gene Phylogenies and Orthologous Groups. In J. C. Setubal, J. Stoye, and P. F. Stadler, editors, *Comparative Genomics: Methods and Protocols*, volume 1704 of *Methods in Molecular Biology*, pages 1–28. Springer New York, 2018. ISBN 978-1-4939-7463-4. doi: 10.1007/978-1-4939-7463-4_1.

[205] L. Shavit, D. Penny, M. D. Hendy, and B. R. Holland. The problem of rooting rapid radiations. *Mol Biol Evol*, 24:2400–2411, 2007. doi: 10.1093/molbev/msm178.

[206] J. M. S. Simões-Pereira. A note on the tree realizability of a distance matrix. *J. Combin. Theory*, 6:303–310, 1969. doi: 10.1016/S0021-9800(69)80092-X.

[207] K. Sjölander, R. S. Datta, Y. Shen, and G. M. Shoffner. Ortholog identification in the presence of domain architecture rearrangement. *Briefings in Bioinformatics*, 12(5): 413–422, June 2011. ISSN 1467-5463. doi: 10.1093/bib/bbr036.

[208] J. Sjöstrand, A. Tofigh, V. Daubin, L. Arvestad, B. Sennblad, and J. Lagergren. A Bayesian Method for Analyzing Lateral Gene Transfer. *Systematic Biology*, 63(3):409–420, Feb. 2014. ISSN 1063-5157. doi: 10.1093/sysbio/syu007.

[209] E. L. Sonnhammer, T. Gabaldón, A. W. Sousa da Silva, M. Martin, M. Robinson-Rechavi, B. Boeckmann, P. D. Thomas, C. Dessimoz, and Q. for Orthologs Consortium. Big data and other challenges in the quest for orthologs. *Bioinformatics*, 30(21):2993–2998, July 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu492.

[210] E. L. L. Sonnhammer and E. V. Koonin. Orthology, paralogy and proposed classification for paralog subtypes. *Trends in Genetics*, 18(12):619–620, Dec. 2002. ISSN 0168-9525. doi: 10.1016/S0168-9525(02)02793-2.

[211] P. F. Stadler, M. Geiß, D. Schaller, A. López Sánchez, M. González Laffitte, D. I. Valdivia, M. Hellmuth, and M. Hernández Rosales. From Best Hits to Best Matches. In *Submitted to: 23th Conference on Algorithmic Computational Biology (RECOMB 2019)*, June 2019.

[212] M. Steel. *Phylogeny: Discrete and Random Processes in Evolution*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, Nov. 2016. ISBN 978-1-61197-447-8.

[213] K. Strimmer and A. von Haeseler. Likelihood-mapping: A simple method to visualize phylogenetic content of a sequence alignment. *Proceedings of the National Academy of Sciences*, 94(13):6815–6819, June 1997. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.94.13.6815.

[214] R. A. Studer and M. Robinson-Rechavi. How confident can we be that orthologs are similar, but paralogs differ? *Trends in Genetics*, 25(5):210–216, 2009. ISSN 0168-9525. doi: https://doi.org/10.1016/j.tig.2009.03.004.

[215] D. P. Sumner. Dacey Graphs. *Journal of the Australian Mathematical Society*, 18(4): 492–502, Dec. 1974. ISSN 0004-9735. doi: 10.1017/S1446788700029232.

[216] 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, 1996.

[217] P. Tabaszewski, P. Górecki, and O. Eulenstein. Phylogenetic Consensus for Exact Median Trees. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '18, pages 366–375. ACM, 2018. ISBN 978-1-4503-5794-4. doi: 10.1145/3233547.3233560.

[218] K. Tamura. Estimation of the number of nucleotide substitutions when there are strong transition-transversion and G+C-content biases. *Molecular Biology and Evolution*, 9(4): 678–687, July 1992. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a040752.

[219] R. E. Tarjan. Applications of path compression on balanced trees. *J. ACM*, 26:690–715, 1979. doi: 10.1145/322154.322161.

[220] R. L. Tatusov, E. V. Koonin, and D. J. Lipman. A Genomic Perspective on Protein Families. *Science*, 278(5338):631–637, Oct. 1997. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.278.5338.631.

[221] 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(1):33–36, Jan. 2000. ISSN 0305-1048. doi: 10.1093/nar/28.1.33.

[222] J. D. Thompson, B. Linard, O. Lecompte, and O. Poch. A Comprehensive Benchmark Study of Multiple Sequence Alignment Methods: Current Challenges and Future Perspectives. *PLOS ONE*, 6(3):e18093, Mar. 2011. ISSN 1932-6203. doi: 10.1371/journal.pone.0018093.

[223] J. W. Thornton and D. B. Kelley. Evolution of the androgen receptor: Structure–function implications. *BioEssays*, 20(10):860–869, 1998. ISSN 1521-1878. doi: 10.1002/(SICI)1521-1878(199810)20:10<860::AID-BIES12>3.0.CO;2-S.

[224] A. Tofigh, M. Hallett, and J. Lagergren. Simultaneous Identification of Duplications and Lateral Gene Transfers. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 8(2): 517–535, Mar. 2011. ISSN 1545-5963. doi: 10.1109/TCBB.2010.14.

[225] 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(14):i75–i82, July 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx229.

[226] 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. doi: 10.1093/bioinformatics/btx229.

[227] B. Vernot, M. Stolzer, A. Goldman, and D. Durand. Reconciliation with non-binary species trees. *J Comput Biol.*, 15:981–1006, 2008. doi: 10.1089/cmb.2008.0092.

[228] D. P. Wall, H. B. Fraser, and A. E. Hirsh. Detecting putative orthologs. *Bioinformatics*, 19(13):1710–1711, Sept. 2003. ISSN 1367-4803. doi: 10.1093/bioinformatics/btg213.

[229] A. Wehe, J. G. Burleigh, and O. Eulenstein. Efficient Algorithms for Knowledge-Enhanced Supertree and Supermatrix Phylogenetic Problems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(6):1432–1441, Nov. 2013. ISSN 1545-5963. doi: 10.1109/TCBB.2012.162.

[230] J. F. Wendel. Genome evolution in polyploids. In J. J. Doyle and B. S. Gaut, editors, *Plant Molecular Evolution*, pages 225–249. Springer Netherlands, Dordrecht, 2000. ISBN 978-94-011-4221-2. doi: 10.1007/978-94-011-4221-2_12.

[231] M. D. Whiteside, G. L. Winsor, M. R. Laird, and F. S. L. Brinkman. OrthologueDB: A bacterial and archaeal orthology resource for improved comparative genomic analysis. *Nucleic Acids Research*, 41(D1):D366–D376, Nov. 2012. ISSN 0305-1048. doi: 10.1093/nar/gks1241.

[232] T. A. Williams, S. E. Heaps, S. Cherlin, T. M. W. Nye, R. J. Boys, and T. M. Embley. New substitution models for rooting phylogenetic trees. *Philos Trans R Soc Lond B Biol Sci*, 370:20140336, 2015. doi: 10.1098/rstb.2014.0336.

[233] Y. I. Wolf and E. V. Koonin. A Tight Link between Orthologs and Bidirectional Best Hits in Bacterial and Archaeal Genomes. *Genome Biology and Evolution*, 4(12):1286–1294, Nov. 2012. ISSN 1759-6653. doi: 10.1093/gbe/evs100.

[234] Z. Yang and B. Rannala. Molecular phylogenetics: Principles and practice. *Nature Reviews Genetics*, 13:303–314, 2012. doi: 10.1038/nrg3186.

[235] M. Yannakakis. Node-and Edge-deletion NP-complete Problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 253–264, New York, NY, USA, 1978. ACM. doi: 10.1145/800133.804355.

[236] C. Yu, N. Zavaljevski, V. Desai, and J. Reifman. QuartetS: A fast and accurate algorithm for large-scale orthology detection. *Nucleic Acids Res*, 39:e88, 2011. doi: 10.1093/nar/gkr308.

[237] C. Yu, V. Desai, L. Cheng, and J. Reifman. QuartetS-DB: A large-scale orthology database for prokaryotes and eukaryotes inferred by evolutionary evidence. *BMC Bioinformatics*, 13(1):143, June 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-143.

[238] B. Zhang and Y.-C. Wu. Coestimation of Gene Trees and Reconciliations Under a Duplication-Loss-Coalescence Model. In Z. Cai, O. Daescu, and M. Li, editors, *Bioinformatics Research and Applications*, pages 196–210. Springer International Publishing, 2017. ISBN 978-3-319-59575-7.

[239] L. Zhang. On a Mirkin-Muchnik-Smith Conjecture for Comparing Molecular Phylogenies. *Journal of Computational Biology*, 4(2):177–187, 1997. doi: 10.1089/cmb.1997.4.177.

[240] Y. Zheng and L. Zhang. Reconciliation With Nonbinary Gene Trees Revisited. *J. ACM*, 64(4):24:1–24:28, Aug. 2017. ISSN 0004-5411. doi: 10.1145/3088512.

[241] E. Zuckerkandl and L. B. Pauling. Molecular disease, evolution, and genic heterogeneity. In M. Kasha and B. Pullman, editors, *Horizons in Biochemistry*, pages 189–225. Academic Press, New York, 1962.

[242] I. E. Zverovich. Near-complete multipartite graphs and forbidden induced subgraphs. *Discrete Mathematics*, 207(1):257–262, Sept. 1999. ISSN 0012-365X. doi: 10.1016/S0012-365X(99)00050-3.

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, den 30.10.2019

Manuela Geiß

Manuela Geiß

Education & Scientific Experience

since Jan 2017 **PhD Studies and Teaching Assistant**, *Bioinformatics Group, Leipzig University*, Germany.

Sept 2016 – Dec 2016 **Research Assistant**, *Institute of Population Genetics, Vetmeduni Vienna*, Austria.

May 2016 – Jul 2016 **Research Assistant**, *Theoretical Biochemistry Group, University of Vienna*, Austria.

Oct 2012 – Apr 2016 **Bachelor Studies in Biology (Genetics and Microbiology)**, *University of Vienna*, Austria, Degree: Bachelor of Science.

Oct 2012 – Nov 2015 **Master Studies in Mathematics**, *University of Vienna*, Austria, Degree: Master of Science (*Graduation with distinction)*.

Oct 2009 – Aug 2012 **Bachelor Studies in Mathematics**, *University of Vienna*, Austria, Degree: Bachelor of Science (*Graduation with distinction)*.

Aug 2000 – Jun 2009 **Secondary School**, *Albert-Einstein Gymnasium*, Schwalbach, Germany, Degree: Abitur (1,0).

Teaching Experience

Mar 2019 – Jul 2019 **Seminars for Algorithms and Datastructures (2)**, *Leipzig University*, Germany.

Oct 2018 – Feb 2019 **Seminars for Algorithms and Datastructures (1)**, *Leipzig University*, Germany.

Mar 2018 – Jul 2018 **Seminars for Algorithms and Datastructures (2)**, *Leipzig University*, Germany.

Oct 2017 – Feb 2018 **Seminars for Algorithms and Datastructures (1)**, *Leipzig University*, Germany.

Mar 2017 – Jul 2017 **Seminars for Algorithms and Datastructures (2)**, *Leipzig University*, Germany.

Languages

German (native), English (fluent), French (fluent)

Awards

2010, 2012, 2013, 2014, 2015 *Leistungsstipendium für hervorragende Studienleistungen (excellence scholarship)*, University of Vienna, Austria.

Publications

Jun 2019 **Complexity of Modification Problems for Reciprocal Best Match Graphs**
M. Hellmuth, M. Geiß, and P.F. Stadler
Submitted to: *Theoretical Computer Science*, arXiv:1907.08865

Jun 2019 **From Best Hits to Best Matches**
P.F. Stadler, M. Geiß, D. Schaller, A. López Sánchez, M. González Laffitte, D.I. Valdivia, M. Hellmuth, and M. Hernández Rosales
Submitted to: *23th Conference on Algorithmic Computational Biology (RECOMB 2019)*

Jun 2019 **Hierarchical Colorings of Cographs**
D.I. Valdivia, M. Geiß, M. Hellmuth, M. Hernández Rosales, and P.F. Stadler
Submitted to: *Discrete Applied Mathematics*, arXiv:1906.10031

Jun 2019 **Best Match Graphs**
M. Geiß, M. González Laffitte, A. López Sánchez, D.I. Valdivia, M. Hellmuth, M. Hernández Rosales, and P.F. Stadler
In: *Journal of Mathematical Biology*, 78, (7), pp. 2015 – 2057, doi:10.1007/s00285-019-01332-9

Apr 2019 **Best Match Graphs and Reconciliation of Gene Trees with Species Trees**
M. Geiß, M. González Laffitte, A. López Sánchez, D.I. Valdivia, M. Hellmuth, M. Hernández Rosales, and P.F. Stadler
Submitted to: *Journal of Mathematical Biology*, arXiv:1904.12021

Mar 2019 **Reciprocal Best Match Graphs**
M. Geiß, P.F. Stadler, and M. Hellmuth
Submitted to: *Journal of Mathematical Biology*, arXiv:1903.07920

Nov 2018 **Reconstructing Gene Trees from Fitch's Xenology Relation**
M. Geiß, J. Anders, P.F. Stadler, Nicolas Wieseke, and M. Hellmuth
In: *Journal of Mathematical Biology*, 77, (5), pp. 1459 – 1491, doi:10.1007/s00285-018-1260-8

Mar 2018 **A Short Note on Undirected Fitch Graphs**
M. Hellmuth, Y. Long, M. Geiß, and P.F. Stadler
In: *The Art of Discrete and Applied Mathematics*, 1, (1), pages #P1.08, doi:10.1007/s00285-018-1260-8

Feb 2018 **Time-Consistent Reconciliation Maps and Forbidden Time Travel**
N. Nøjgaard, M. Geiß, D. Merkle, P.F. Stadler, N. Wieseke, and M. Hellmuth
In: *Algorithms for Molecular Biology*, 13:2, doi:10.1186/s13015-018-0121-8

Aug 2017 **Forbidden Time Travel: Characterization of Time-Consistent Tree Reconciliation Maps**
N. Nøjgaard, M. Geiß, D. Merkle, P.F. Stadler, N. Wieseke, and M. Hellmuth
In: *17th International Workshop on Algorithms in Bioinformatics (WABI 2017)*, 88, pp. 17:1–17:12, doi:10.4230/LIPIcs.WABI.2017.17