Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 25, no. 1, pp. 397–416 (2021)
DOI: 10.7155/jgaa.00564

Least resolved trees for two-colored best match graphs

David Schaller[1,2] *Manuela Geiß*[3] *Marc Hellmuth*[4] *Peter F. Stadler*[1,2,5,6,7]

[1]Max Planck Institute for Mathematics in the Sciences, Inselstraße 22, D-04103 Leipzig, Germany
[2]Bioinformatics Group, Department of Computer Science & Interdisciplinary Center for Bioinformatics, Universität Leipzig, Härtelstraße 16–18, D-04107 Leipzig, Germany
[3]Software Competence Center Hagenberg GmbH, Softwarepark 21, A-4232 Hagenberg, Austria
[4] Department of Mathematics, Faculty of Science, Stockholm University, SE - 106 91 Stockholm, Sweden
[5]Institute for Theoretical Chemistry, University of Vienna, Währingerstraße 17, A-1090 Wien, Austria
[6]Facultad de Ciencias, Universidad Nacional de Colombia, Bogotá, Colombia
[7]The Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501, USA

Submitted: January 2021	Reviewed: June 2021	Revised: June 2021
Accepted: July 2021	Final: July 2021	Published: September 2021
Article type: Regular paper	Communicated by: F. Vandin	

Abstract. In phylogenetic combinatorics, 2-colored best match graphs (2-BMGs) form a subclass of sink-free bi-transitive digraphs that describe the most closely related genes between a pair of species in an evolutionary scenario. They are explained by a unique least resolved tree (LRT). In this paper, the concept of support vertices is introduced and used to derive an $O(|V|+|E|\log^2|V|)$-time algorithm that recognizes a 2-BMG and constructs its LRT. The approach can be extended to allow the recognition of binary-explainable 2-BMGs with the same complexity. An empirical comparison emphasizes the efficiency of the new algorithm.

1 Introduction

Best match graphs (BMGs) have been introduced recently in phylogenetic combinatorics to formalize the notion of a gene y in species 2 being an evolutionary closest relative of a gene x in species 1, i.e., y is a best match for x [6]. The best matches between genes of two species form a bipartite

This work was supported in part by the Austrian Federal Ministries BMK and BMDW and the Province of Upper Austria in the frame of the COMET Programme managed by FFG, and the *Deutsche Forschungsgemeinschaft* proj. no. STA850/49-1.

E-mail addresses: sdavid@bioinf.uni-leipzig.de (David Schaller) manuela.geiss@scch.at (Manuela Geiß) marc.hellmuth@math.su.se (Marc Hellmuth) studla@bioinf.uni-leipzig.de (Peter F. Stadler)

directed graph (digraph), the 2-colored best match graph or 2-BMG, whose structure is determined by the phylogenetic tree describing the evolution of the genes. 2-BMGs are characterized by four local properties [6, 11] that relate them to classes of digraphs which appear of interest also from a theoretical point of view. A bipartite digraph $\vec{G} = (V, E)$ is a 2-BMG if it satisfies

(N0) Every vertex has at least one out-neighbor, i.e., $\vec{G}$ is *sink-free*.

(N1) If u and v are two independent vertices, then there exist no vertices w and t such that $(u, t), (v, w), (t, w) \in E$.

(N2) For any four vertices u_1, u_2, v_1, v_2 with $(u_1, v_1), (v_1, u_2), (u_2, v_2) \in E$ we have $(u_1, v_2) \in E$, i.e., $\vec{G}$ is *bi-transitive*.

(N3) For any two vertices u and v with a common out-neighbor, if there exists no vertex w such that either $(u, w), (w, v) \in E$, or $(v, w), (w, u) \in E$, then u and v have the same in-neighbors and either all out-neighbors of u are also out-neighbors of v or all out-neighbors of v are also out-neighbors of u.

Sink-free digraphs have appeared in particular in the context of graph semigroups [1] and graph orientation problems [3]. Bi-transitive graphs were introduced in [5] in the context of oriented bipartite graphs and investigated in relation with topological orderings in [10, 11]. The class of digraphs satisfying (N1), (N2), and (N3) are characterized by a system of forbidden induced subgraphs [15], see Theorem 2.

In [6], best match graphs are defined as vertex-colored digraphs $(\vec{G}, \sigma)$, where the vertex coloring σ assigns to each gene x the species $\sigma(x)$ in which it resides. The subgraphs of a BMG induced by vertices of two distinct colors form a 2-BMG. In this context, the vertex coloring is assigned *a priori*, while the definition above induces a coloring that is unique only up to relabeling the colors independently on each (weakly) connected component of $\vec{G}$. Throughout this contribution, we will view BMGs as properly vertex-colored digraphs $(\vec{G}, \sigma)$, see Definition 1 below.

For each BMG $(\vec{G}, \sigma)$, there is a unique least resolved leaf-colored tree (T^*, σ) with leaves corresponding to the vertices of $(\vec{G}, \sigma)$ such that the arcs in $(\vec{G}, \sigma)$ are the best matches w.r.t. (T^*, σ) (cf. Definition 1). Figure 1 shows an example of a 2-BMG together with its least resolved tree. Using certain sets of rooted triples that can be inferred from the 2-colored induced subgraphs of $(\vec{G}, \sigma)$ with three vertices, it is possible to determine whether $(\vec{G}, \sigma)$ is a BMG in polynomial time and, if so, to construct the least resolved tree (T^*, σ) [6, 12]. These papers also describe $O(|V|^3)$-time algorithms for the recognition of 2-BMGs and the construction of the LRT for a given 2-BMG.

In the present contribution, we obtain a new characterization of 2-BMGs that avoids the use of rooted triples. This will give rise to an alternative, efficient algorithm for the recognition of 2-BMGs and the construction of the least resolved trees. The paper is organized as follows: In Section 2, we introduce the necessary notation and review some results from the literature. Section 3 is concerned with a more detailed analysis of the least resolved trees (LRTs) of BMGs with an arbitrary number of colors. We then turn to the peculiar properties of the LRTs of 2-BMGs in Section 4. To this end, we introduce the concept of "support leaves" that uniquely determine LRTs. The main result of this section is Theorem 3, which shows that the support leaves of the root can be identified directly in a 2-BMG. In Section 5, we then turn Theorem 3 into an efficient algorithm for recognizing 2-BMGs and constructing their LRTs. Computational experiments demonstrate the performance gain in practice. In Section 6 we extend the algorithmic approach to binary-explainable 2-BMGs, a subclass that features an additional forbidden induced subgraph.

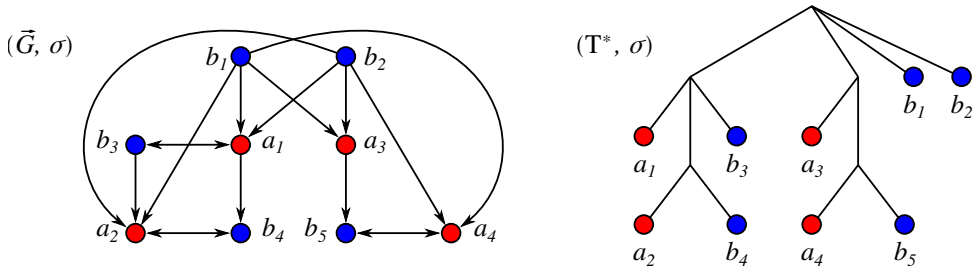

Figure 1: Example of a 2-BMG $(\vec{G}, \sigma)$ and its explaining least resolved tree (T^*, σ).

2 Preliminaries

Let $T = (V, E)$ be a tree with root ρ and leaf set $L \coloneqq L(T) \subset V$. The set of inner vertices of T is $V^0(T) \coloneqq V \setminus L$, in particular ρ is an inner vertex. An edge $e = uv \in E$ is an *inner* edge of T if u and v are both inner vertices. Otherwise it is an *outer* edge. A vertex $u \in V$ is an *ancestor* of $v \in V$ in T if u lies on the path from ρ to v. In this case, we write $v \preceq_T u$. For an edge $uv \in E$, we use the convention that $v \prec_T u$. A vertex v is a *child* of u if $uv \in E$ and $v \prec_T u$. We write $\mathrm{child}_T(u)$ for the set of children of u in T. The *least common ancestor* $\mathrm{lca}_T(A)$ is the unique $\preceq_T$-smallest vertex that is an ancestor of all vertices in $A \subseteq V$. For brevity, we write $\mathrm{lca}_T(x, y)$ instead of $\mathrm{lca}_T(\{x, y\})$. The subtree of T rooted in a vertex $u \in V$ is denoted by $T(u)$. A *leaf coloring* of a tree is a map $\sigma\colon L \to M$ where M is a non-empty set of *colors*. We consider leaf-colored trees (T, σ) and write $\sigma(L') \coloneqq \{\sigma(v) \mid v \in L'\}$ for subsets $L' \subseteq L$.

Definition 1 *Let (T, σ) be a leaf-colored tree. A leaf $y \in L(T)$ is a* best match *of the leaf $x \in L(T)$ if $\sigma(x) \neq \sigma(y)$ and $\mathrm{lca}(x, y) \preceq_T \mathrm{lca}(x, y')$ holds for all leaves y' of color $\sigma(y') = \sigma(y)$.*

Given (T, σ), the digraph $\vec{G}(T, \sigma) = (V, E)$ with vertex set $V = L(T)$, vertex coloring σ, and with arc $(x, y) \in E$ if and only if y is a best match of x w.r.t. (T, σ) is the *best match graph* (BMG) of (T, σ) [6].

Definition 2 *An arbitrary vertex-colored digraph $(\vec{G}, \sigma)$ is a* best match graph (BMG) *if there exists a leaf-colored tree (T, σ) such that $(\vec{G}, \sigma) = \vec{G}(T, \sigma)$. In this case, we say that (T, σ)* explains *$(\vec{G}, \sigma)$.*

We say that $(\vec{G}, \sigma)$ is an ℓ-BMG if $\sigma : V(\vec{G}) \to S$ is surjective and $|S| = \ell > 0$. Given a directed graph $\vec{G} = (V, E)$ we denote the set of out-neighbors of a vertex $x \in V$ by $N^+(x) \coloneqq \{y \in V | (x, y) \in E(\vec{G})\}$ and the out-degree $|N^+(x)|$ of x by $\mathrm{outdeg}(x)$. Similarly, $N^-(x) \coloneqq \{y \in V | (y, x) \in E(\vec{G})\}$ denotes the set of in-neighbors of x. By construction, the coloring σ of a BMG $(\vec{G}, \sigma)$ is *proper*, i.e., $x \in N^+(y)$ implies $\sigma(x) \neq \sigma(y)$, and there is at least one best match of x for every color $s \in \sigma(V) \setminus \{\sigma(x)\}$. In particular, therefore, we have $N^+(x) \neq \emptyset$ for every 2-BMG, i.e., every 2-BMG is sink-free. Note that BMGs will in general have sources, i.e., $N^-(x)$ may be empty. We write $\vec{G}[W]$ for the subgraph of $\vec{G} = (V, E)$ induced by $W \subseteq V$ and $\vec{G} - W$ for $\vec{G}[V \setminus W]$. A directed graph is (weakly) connected if its underling undirected graph is connected. A *connected component* is a maximal connected subgraph of $\vec{G}$.

Following [16], we say that T' is *displayed* by T, in symbols $T' \leq T$, if the tree T' can be obtained from a subtree of T by contraction of edges. For leaf-colored trees, we say that (T,σ) *displays* or *is a refinement of* (T',σ'), whenever $T' \leq T$ and $\sigma(v) = \sigma'(v)$ for all $v \in L(T')$.

Definition 3 *A leaf-colored trees (T,σ) is least resolved with respect to $\vec{G}(T,\sigma)$ if there is no tree (T',σ) such that $T' < T$, i.e., $T' \leq T$ and $T' \neq T$, which also explains $\vec{G}(T,\sigma)$.*

Theorem 1 ([6], Theorem 9**)** *If $(\vec{G},\sigma)$ is a BMG, then there is a unique least resolved tree (T,σ) that explains $(\vec{G},\sigma)$.*

The concept of least resolved trees (LRTs) is closely related to that of redundant edges.

Definition 4 *An edge $e \in E(T)$ is redundant with respect to $\vec{G}(T,\sigma)$ if the tree T_e obtained by contracting the edge e satisfies $\vec{G}(T_e,\sigma) = \vec{G}(T,\sigma)$.*

A tree (T,σ) is least resolved if and only if it does not contain redundant edges (cf. Theorem 8 in [6]). In the following, we will make use of the following characterization of redundant edges.

Lemma 1 ([14], Lemma 2.10**)** *Let $(\vec{G},\sigma)$ be a BMG explained by a tree (T,σ). The edge $e = uv$ in (T,σ) is redundant w.r.t. $(\vec{G},\sigma)$ if and only if (i) e is an inner edge of T and (ii) there is no arc $(a,b) \in E(\vec{G})$ such that $\mathrm{lca}_T(a,b) = v$ and $\sigma(b) \in \sigma(L(T(u)) \setminus L(T(v)))$.*

In the following, we will frequently need to restrict the coloring σ on $\vec{G}$ or $L(T)$ to a subset of vertices or leaves. Since, in situations like $(G_i, \sigma_{|V(G_i)})$, the set to which σ is restricted is clear, we will write $\sigma_{|\cdot}$ to keep the notation concise.

BMGs can also be described in terms of their connected components.

Proposition 1 ([6], Proposition 1**)** *A digraph $(\vec{G},\sigma)$ is an ℓ-BMG if and only if all its connected components are ℓ-BMGs.*

As a simple consequence of Proposition 1 and the definition of ℓ-BMGs, all connected components $(G_i,\sigma_{|\cdot})$ and $(G_j,\sigma_{|\cdot})$ of an ℓ-BMG satisfy $\sigma(V(G_i)) = \sigma(V(G_j))$ and $|\sigma(V(G_j))| = \ell$. For our purposes it will also be important to relate the structure of a tree (T,σ) to the connectedness of the BMG $\vec{G}(T,\sigma)$.

Proposition 2 ([6], Theorem 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 such that $\sigma(L(T(v))) \neq \sigma(L(T))$. Furthermore, if $\vec{G}(T,\sigma)$ is not connected, then for every connected component $\vec{G}_i$ of $\vec{G}(T,\sigma)$ there is a child v of the root ρ such that $V(G_i) \subseteq L(T(v))$.*

Moreover, 2-BMGs can be characterized by three types of forbidden subgraphs [15]. To this end, we will need the following classes of small bipartite digraphs.

Definition 5 (F1-, F2-, and F3-graphs)

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

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

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

Theorem 2 ([15], Theorem 3.4) *A properly 2-colored digraph is a 2-BMG if and only if it is sink-free and does not contain an induced F1-, F2-, or F3-graph.*

As noted in [15], the forbidden induced F1-, F2-, and F3-subgraphs characterize exactly the class of bipartite directed graphs satisfying the Axioms (N1), (N2), and (N3) mentioned in the introduction.

Although we aim at avoiding the use of triples in the final results, we will need them during our discussion. A triple $ab|c$ is a rooted tree t on three pairwise distinct vertices $\{a, b, c\}$ such that $\mathrm{lca}_t(a, b) \prec_t \mathrm{lca}_t(a, c) = \mathrm{lca}_t(b, c) = \rho$, where ρ denotes the root of t. A set $\mathcal{R}$ of triples is *consistent* if there is a tree T that displays all triples in $\mathcal{R}$. Given a vertex-colored digraph $(\vec{G}, \sigma)$, we define its set of *informative triples* [6, 14] as

$$\mathcal{R}(\vec{G}, \sigma) := \left\{ ab|b' \colon \sigma(a) \neq \sigma(b) = \sigma(b'),\ (a, b) \in E(\vec{G}); (a, b') \notin E(\vec{G}) \right\}. \tag{1}$$

Lemma 2 ([14], Lemmas 2.8 and 2.9) *If* $(\vec{G}, \sigma)$ *is a BMG, then every tree* (T, σ) *that explains* $(\vec{G}, \sigma)$ *displays all triples* $t \in \mathcal{R}(\vec{G}, \sigma)$.
Moreover, if the triples $ab|b'$ *and* $cb'|b$ *are informative for* $(\vec{G}, \sigma)$*, then every tree* (T, σ) *that explains* $(\vec{G}, \sigma)$ *contains two distinct children* $v_1, v_2 \in \mathrm{child}_T(\mathrm{lca}_T(a, c))$ *such that* $a, b \prec_T v_1$ *and* $b', c \prec_T v_2$.

Observation 1 *Let* (T, σ) *be a tree explaining the BMG* $(\vec{G}, \sigma)$*, and* $v \in V(T)$ *a vertex such that* $\sigma(L(T(v))) = \sigma(L(T))$. *Then* $(a, b) \in E(\vec{G})$ *and* $a \in L(T(v))$ *implies* $b \in L(T(v))$.

Finally, there is a close connection between subtrees of T and subgraphs of $\vec{G}(T, \sigma)$.

Lemma 3 ([12], Lemmas 22 and 23) *Let* (T, σ) *be a tree explaining a BMG* $(\vec{G}, \sigma)$. *Then* $\vec{G}(T(u), \sigma_{|.}) = (\vec{G}[L(T(u))], \sigma_{|.})$ *holds for every* $u \in V(T)$. *Moreover, if* (T, σ) *is least resolved for* $(\vec{G}, \sigma)$*, then the subtree* $T(u)$ *is least resolved for* $\vec{G}(T(u), \sigma_{|.})$.

3 Properties of Least Resolved Trees

In this section, we derive some helpful properties of LRTs which we will use repeatedly throughout this work.

Lemma 4 *Let* $(\vec{G}, \sigma)$ *be a BMG and* (T, σ) *its least resolved tree. Then the BMG* $\vec{G}(T(v), \sigma_{|.})$ *is connected for every* $v \in V(T)$ *with* $v \prec_T \rho_T$.

Proof: By Lemma 3, $\vec{G}(T(v), \sigma_{|.})$ is a BMG. First observe that $\vec{G}(T(v), \sigma_{|.})$ is trivially connected if v is a leaf. Now let $v \prec_T \rho_T$ be an arbitrary inner vertex of T. Thus, there exists a vertex $u \succ_T v$ such that uv is an inner edge. Since (T, σ) is least resolved, it does not contain any redundant edges. Hence, since uv is an inner edge, there is an arc $(a, b) \in E(\vec{G})$ such that $\mathrm{lca}_T(a, b) = v$ and $\sigma(b) \in \sigma(L(T(u)) \setminus L(T(v)))$ by Lemma 1. Since $a, b \in L(T(v))$, Lemma 3 implies that (a, b) is also an arc in $\vec{G}(T(v), \sigma_{|.})$. Moreover, $\mathrm{lca}_{T(v)}(a, b) = v$ clearly also holds in the subtree rooted at v. Now consider the child $w \in \mathrm{child}_{T(v)}(v)$ such that $a \preceq_{T(v)} w$. There cannot be a leaf $b' \in L(T(w))$

with $\sigma(b') = \sigma(b)$ since otherwise $\mathrm{lca}_{T(v)}(a, b') \preceq_{T(v)} w \prec_{T(v)} v$ would contradict that (a, b) is an arc in $\vec{G}(T(v), \sigma_{|.})$. Thus $\sigma(b) \notin \sigma(L(T(w)))$. Since $\sigma(b) \in \sigma(L(T(v)))$, we thus conclude $\sigma(L(T(w))) \neq \sigma(L(T(v)))$. The latter together with Proposition 2 implies that $\vec{G}(T(v), \sigma_{|.})$ is connected. □

The converse of Lemma 4, however, is not true, i.e., a tree (T, σ) for which $\vec{G}(T(v), \sigma_{|.})$ is connected for every $v \in V(T)$ with $v \prec_T \rho_T$ is not necessarily least resolved. To see this, consider the caterpillar tree (T, σ) given by $(x'', (x', (x, y)))$ with $\sigma(x) = \sigma(x') = \sigma(x'') \neq \sigma(y)$ and $u = \mathrm{lca}_T(x, x')$. It is straightforward to verify that the BMG of each subtree of T is connected. However, the edge $\rho_T u$ is redundant.

Lemma 5 *Let (T, σ) be the least resolved tree of some BMG $(\vec{G}, \sigma)$. Then every vertex $v \prec_T \rho_T$ with $|\sigma(L(T(v)))| = 1$ is a leaf.*

Proof: Let $v \prec_T \rho_T$ with $|\sigma(L(T(v)))| = 1$ and assume, for contradiction, that v is not a leaf. Hence, $|L(T(v))| > 1$. By Lemma 3 $\vec{G}(T(v), \sigma_{|.})$ is a BMG and, therefore, properly colored. But then $\vec{G}(T(v), \sigma_{|.})$ is disconnected; a contradiction to Lemma 4. □

An immediate consequence of Lemma 5 is the following corollary.

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

Proof: If $|\sigma(L(T(v)))| = 1$, Lemma 5 implies that v is a leaf. Otherwise, if $|\sigma(L(T(v)))| > 1$, $T(v)$ clearly must contain at least two leaves and thus v cannot be a leaf. □

4 Support Leaves

In this section, we introduce "support leaves" to recursively construct the LRT of a 2-BMG. The main result of this section shows that these leaves can be inferred directly from a BMG without any further knowledge of its corresponding LRT. We start with a technical result similar to Corollary 3 in [6]; here we use a much simpler and more convenient notation.

Lemma 6 *Let (T, σ) be the least resolved tree of a 2-BMG $(\vec{G}, \sigma)$. Then, for every vertex $u \in V^0(T) \setminus \{\rho_T\}$, $\mathrm{child}_T(u) \cap L(T) \neq \emptyset$. If $(\vec{G}, \sigma)$ is connected, then $\mathrm{child}_T(u) \cap L(T) \neq \emptyset$ holds for* every *$u \in V^0(T)$.*

Proof: First, suppose that $(\vec{G}, \sigma)$ is disconnected and let $u \in V^0(T) \setminus \{\rho_T\}$. Since (T, σ) is least resolved, Lemma 4 implies that $\vec{G}(T(u), \sigma_{|.})$ is connected for every $u \in V(T)$ with $u \prec_T \rho_T$. Hence, we can apply Proposition 2 to $\vec{G}(T(u), \sigma_{|.})$ and conclude that there is a child $v \in \mathrm{child}_{T(u)}(u)$ such that $\sigma(L(T(v))) \subsetneq \sigma(L(T(u)))$. Since (T, σ) is 2-colored, the latter immediately implies $|\sigma(L(T(v)))| = 1$ and, by Lemma 5, v is a leaf. Thus every $u \in V^0(T) \setminus \{\rho_T\}$ has a leaf v among its children, i.e. $\mathrm{child}_T(u) \cap L(T) \neq \emptyset$. If $(\vec{G}, \sigma)$ is connected, we can apply the same argument to $u = \rho_T$ and conclude that a leaf v is attached to ρ_T. □

Lemma 6 states that, in the least resolved tree of a connected 2-BMG, every inner vertex u is adjacent to at least one leaf, and thus in a way "supported" by it.

Definition 6 (Support Leaves) *For a given tree T, the set $S_u := \mathrm{child}_T(u) \cap L(T)$ is the set of all* support leaves *of vertex $u \in V(T)$.*

Note that Lemma 6 is in general not true for ℓ-BMGs with $\ell \geq 3$. For example, consider the (least resolved) tree $((a,b),(c,a'))$ with three distinct leaf colors $\sigma(a) = \sigma(a') \neq \sigma(b) \neq \sigma(c)$.

A straightforward consequence of Proposition 2 and Lemma 5 is given in the following corollary.

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

Proof: By Proposition 2, $\vec{G}(T,\sigma)$ is connected if and only if there exists a child v of the root ρ of T such that $T(v)$ does not contain all colors. Thus $|\sigma(L(T(v)))| = 1$. By Lemma 5, we have $|\sigma(L(T(v)))| = 1$ if and only if v is a leaf, i.e. $v \in S_\rho$. Hence, $\vec{G}(T,\sigma)$ is connected if and only if $S_\rho \neq \emptyset$. □

Lemma 7 *Let (T,σ) be the least resolved tree of a 2-BMG $(\vec{G},\sigma)$, and S_ρ the set of support leaves of the root ρ. Then the connected components of $(\vec{G} - S_\rho, \sigma_{|.})$ are exactly the 2-BMGs $\vec{G}(T(v),\sigma_{|.})$ with $v \in \mathrm{child}(\rho) \setminus S_\rho$.*

Proof: Let $v \in \mathrm{child}_T(\rho) \cap V^0(T) = \mathrm{child}_T(\rho) \setminus S_\rho$ and consider the 2-BMG $\vec{G}(T(v),\sigma_{|.})$. By Lemmas 4 and 3, $\vec{G}(T(v),\sigma_{|.})$ is connected and we have $\vec{G}(T(v),\sigma_{|.}) = (\vec{G}[L(T(v))],\sigma_{|.})$, respectively. Moreover, it holds $((\vec{G} - S_\rho)[L(T(v))],\sigma_{|.}) = (\vec{G}[L(T(v))],\sigma_{|.})$ since $L(T(v)) = V(\vec{G}[L(T(v))]) = V(H[L(T(v))])$ for $H \coloneqq \vec{G} - S_\rho = \vec{G}[V(\vec{G}) \setminus S_\rho]$.

If $\mathrm{child}_T(\rho) \setminus S_\rho = \{v\}$, then the statement is trivially satisfied. Therefore, suppose that $|\mathrm{child}_T(\rho) \setminus S_\rho| > 1$. Hence, it remains to show that there are no arcs between $\vec{G}(T(v),\sigma_{|.})$ and $\vec{G}(T(w),\sigma_{|.})$ for any $w \in \mathrm{child}_T(\rho) \setminus S_\rho$, $w \neq v$. Corollary 1 and $v \prec_T \rho$ imply that $T(v)$ contains both colors. Thus, by Observation 1, there are no out-arcs from $L(T(v))$ to any vertex in $L(T) \setminus L(T(v))$, hence in particular there are no out-arcs (x,y) with $x \preceq_T v$, $y \preceq_T w$. The same holds for w, thus we can conclude that there are no arcs (y,x). From the observation that each $x \in L(T) \setminus S_\rho$ must be located below some $v \in \mathrm{child}_T(\rho) \cap V^0(T)$, it now immediately follows that $(\vec{G} - S_\rho, \sigma_{|.})$ consists exactly of these connected components as stated. □

As a consequence, we have

Corollary 3 *Let (T,σ) with root ρ be the LRT of a 2-BMG $(\vec{G},\sigma)$. Then each child of ρ is either one of the support leaves S_ρ of ρ or the root of the LRT for a connected component of $(\vec{G} - S_\rho, \sigma_{|.})$.*

Proof: The support leaves S_ρ are children of ρ by definition. By Lemma 7, the connected components of $(\vec{G} - S_\rho, \sigma_{|.})$ are exactly the 2-BMGs $\vec{G}(T(v),\sigma_{|.})$ with $v \in \mathrm{child}_T(\rho) \setminus S_\rho$. Moreover, by Lemma 3, the subtrees $T(v)$ with $v \in \mathrm{child}_T(\rho) \setminus S_\rho$ are exactly the unique LRTs for these 2-BMGs. □

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

Definition 7 (Umbrella Vertices) *For an arbitrary properly 2-colored digraph $(\vec{G},\sigma)$, the set*

$$U(\vec{G},\sigma) \coloneqq \left\{ x \in V(\vec{G}) \mid y \in V(\vec{G}) \text{ and } \sigma(y) \neq \sigma(x) \implies y \in N^+(x) \right\}$$

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

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

Definition 8 (Support Set of $(\vec{G},\sigma)$) *Let $(\vec{G},\sigma)$ be a properly 2-colored digraph. A* support set $S := S(\vec{G},\sigma)$ *of $(\vec{G},\sigma)$ is a maximal subset $S \subseteq U(\vec{G},\sigma)$ of umbrella vertices such that $x \in S$ implies $N^-(x) \subseteq S$.*

Lemma 8 *Every properly 2-colored digraph $(\vec{G},\sigma)$ has a unique support set $S(\vec{G},\sigma)$.*

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

In order to construct the support set S, we consider the following sequence of sets, defined recursively by

$$S^{(k)} := \{x \in S^{(k-1)} \mid N^-(x) \subseteq S^{(k-1)}\} \text{ for } k \geq 1 \text{ and } S^{(0)} = U(\vec{G},\sigma). \tag{2}$$

By construction $S^{(k+1)} \subseteq S^{(k)}$ and thus there is a non-negative integer $k < |V(\vec{G})|$ such that $S^{(k+1)} = S^{(k)}$. In this case, we have $S^{(j)} = S^{(k)}$ for all $j > k$. Thus it follows directly from the definition that $S = S^{(k)}$ if and only if $S^{(k+1)} = S^{(k)}$. The following result shows that S is obtained in a single iteration whenever $(\vec{G},\sigma)$ is a 2-BMG.

Lemma 9 *If $(\vec{G},\sigma)$ is a 2-BMG, then $S = S^{(1)}$.*

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

As an immediate consequence, a 2-BMG satisfies $S^{(1)} = S^{(2)}$. This condition will be used in Algorithm 1 below. On the other hand, 2-BMGs in general do not satisfy $S = S^{(0)} = U(\vec{G},\sigma)$. To see this, consider the BMG $(\vec{G},\sigma)$ that is explained by the triple $x_1y|x_2$ with $\sigma(x_1) = \sigma(x_2) \neq \sigma(y)$. One easily verifies that $U(\vec{G},\sigma) = \{x_1,x_2\}$ but $S = \{x_2\}$.

Theorem 3 *Let (T,σ) be the least resolved tree of a 2-BMG $(\vec{G},\sigma)$. Then, the set of support leaves S_ρ of the root ρ equals the support set S of $(\vec{G},\sigma)$. In particular, $S \neq \emptyset$ if and only if $(\vec{G},\sigma)$ is connected.*

Proof: Let (T,σ) be the LRT of a 2-BMG $(\vec{G}=(V,E),\sigma)$. We set $U \coloneqq U(\vec{G},\sigma)$. Note that $S = S^{(1)}$ by Lemma 9.

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

Now, suppose that $(\vec{G},\sigma)$ is connected. By Corollary 2, we have $S_\rho \neq \emptyset$. We first show $S_\rho \subseteq S$. Let $x \in S_\rho$. By definition, x satisfies $\mathrm{lca}_T(x,y) = \rho$ and therefore $(x,y) \in E$ for all $y \in L(T)$ with $\sigma(y) \neq \sigma(x)$, i.e., x has an out-arc to every differently colored vertex in $\vec{G}$. By definition, we thus have $x \in U$. Now assume for contradiction that $x \notin S = S^{(1)} = \{z \in U \mid N^-(z) \subseteq U\}$. The latter implies that there exists a vertex $y \in N^-(x)$ such that $y \notin U$. In particular, $(y,x) \in E$. Since $y \notin U$, there is some vertex x' with $\sigma(x') = \sigma(x)$ such that $(y,x') \notin E$. This implies that $xy|x'$ is an informative triple. By Lemma 2, we obtain $\mathrm{lca}_T(x,y) \prec_T \mathrm{lca}_T(x,x') = \mathrm{lca}_T(x',y) \preceq_T \rho$; contradicting the assumption that x is a support leaf of ρ. Thus $x \in S$.

Next, we show that $S \subseteq S_\rho$. Suppose that $x \in S$ is not a support leaf of ρ, i.e. $x \notin S_\rho$. Hence, there is an inner vertex $v \in \mathrm{child}_T(\rho) \cap V^0(T)$ such that $x \prec_T v$. By Corollary 1, we conclude that $|\sigma(L(T(v)))| = 2$, i.e., the subtree $T(v)$ contains both colors. We now distinguish two cases: either (i) there is a leaf $y' \in L(T) \setminus L(T(v))$ with $\sigma(y') \neq \sigma(x)$, or (ii) there is no leaf $y' \in L(T) \setminus L(T(v))$ with $\sigma(y') \neq \sigma(x)$.

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

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

In summary, we have shown $S = S_\rho$ for any 2-BMG $(\vec{G},\sigma)$. Finally, $S = S_\rho$ together with Corollary 2 implies that $S \neq \emptyset$ if and only if $(\vec{G},\sigma)$ is connected, which completes the proof. □

5 Algorithmic Considerations

Theorem 3 provides not only a convenient necessary condition for connected 2-BMGs, but also a fast way of determining the support set $S = S_\rho$, and thus also a fast recursive approach to construct the LRT of a 2-BMG which is formalized in Algorithm 1 and illustrated in Figure 2.

Algorithm 1: LRT for connected 2-BMGs $(\vec{G}, \sigma)$.

Input: Connected properly 2-colored digraph $(\vec{G} = (L, E), \sigma)$, vertex ρ
Output: LRT of $(\vec{G}, \sigma)$ if $(\vec{G}, \sigma)$ is a BMG

```
1  Function Build2ColLRT(G⃗, σ, ρ)
2      U ← {x ∈ L | outdeg(x) = |L| − |L[σ(x)]|}          // umbrella vertices
3      S^(1) ← {x ∈ U | N⁻(x) ⊆ U}                        // all in-neighbors in U
4      S^(2) ← {x ∈ S^(1) | N⁻(x) ⊆ S^(1)}                // all in-neighbors in S^(1)
5      if S^(1) = ∅ or S^(2) ≠ S^(1) then
6          exit false                                      // not a 2-BMG
7      else
8          foreach x ∈ S^(2) do
9              add x as a child of ρ
10         foreach connected component G⃗_v of G⃗ − S^(2) do
11             if |V(G⃗_v)| = 1 then
12                 exit false                              // not a 2-BMG
13             create vertex v
14             T_v ←Build2ColLRT(G⃗_v, σ|., v)
15             connect the root v of T_v as a child to ρ
```

Lemma 10 *Let $(\vec{G}, \sigma)$ be a connected 2-BMG. Then Algorithm 1 returns the least resolved tree for $(\vec{G}, \sigma)$.*

Proof: Let (T, σ) be the (unique) least resolved tree of $(\vec{G}, \sigma)$ with root ρ. The latter is supplied to Algorithm 1 to initialize the tree. Since $(\vec{G}, \sigma)$ is connected, Theorem 3 and Lemma 9 imply that the set of support leaves $S_\rho = S^{(2)} = S^{(1)} \neq \emptyset$ for the root ρ is correctly identified in the top-level recursion of Algorithm 1 (Line 2-4) and attached to the root ρ (Line 8-9). According to Corollary 3, one can now proceed to recursively construct the LRTs for the connected components of $(\vec{G} - S_\rho, \sigma_{|.})$, which is done in Line 10-15. By Lemma 7, these connected components $(\vec{G}_v, \sigma_{|.})$ are exactly the BMGs $\vec{G}(T(v), \sigma_{|.})$ with $v \in \mathrm{child}_T(\rho) \setminus \{S_\rho\}$ (Line 14). In particular, therefore, we have $V(\vec{G}_v) = L(T(v))$. Since $v \notin S_\rho$, i.e., v is an inner vertex, Corollary 1 and $v \prec_T \rho$ imply $|\sigma(L(T(v)))| > 1$. Hence, in particular, the condition $|V(\vec{G}_v)| > 1$ (cf. Line 11) to proceed recursively is satisfied for each connected component. □

Theorem 4 *Given a connected properly 2-colored digraph $(\vec{G}, \sigma)$, Algorithm 1 returns a tree T if and only if $(\vec{G}, \sigma)$ is a 2-BMG. In particular, T is the unique least resolved tree for $(\vec{G}, \sigma)$.*

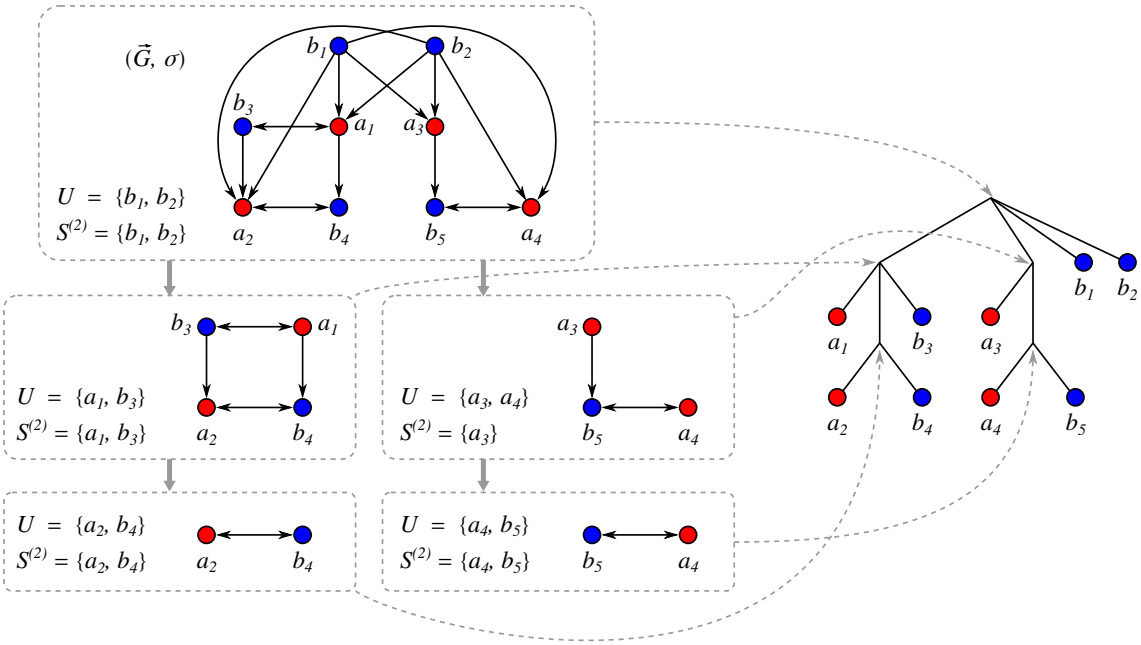

Figure 2: Illustration of Algorithm 1 with input $(\vec{G}, \sigma)$ (uppermost box). The boxes indicate the five recursion steps that are necessary to decompose $(\vec{G}, \sigma)$, and correspond to the five inner vertices of the LRT shown on the right. Note that, in the recursion step on $(\vec{G}[\{a_3, a_4, b_5\}], \sigma_{|.})$, we have $U \neq S^{(2)}$.

Proof: By Lemma 10, Algorithm 1 returns the unique least resolved tree T if $(\vec{G}, \sigma)$ is a connected 2-BMG. To prove the converse, suppose that Algorithm 1 returns a tree T given the connected properly 2-colored digraph $(\vec{G}, \sigma)$ as input. We will show that $(\vec{G}, \sigma) = \vec{G}(T, \sigma)$, and thus $(\vec{G}, \sigma)$ is a BMG.

It is straightforward to see that $L(T) = V(\vec{G})$ must hold since, in each step of Algorithm 1 every vertex is either attached to some inner vertex or passed down to a deeper-level recursion as part of some connected component. Therefore, every vertex of $\vec{G}$ eventually appears in the output. Thus $\sigma(L(T)) = \sigma(V(\vec{G}))$ and $|\sigma(L(T))| = |\sigma(V(\vec{G}))| = 2$. It remains to show $E(\vec{G}) = E(\vec{G}(T, \sigma))$.

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

First, let S be the support set of $\vec{G}(T, \sigma)$ and $x \in S$ be arbitrary. Note that the support set is computed in the first iteration step of the algorithm as $S = S_1^{(2)}$, hence $S = S_1^{(2)} \neq \emptyset$. By construction of T, x is attached as a leaf to ρ, i.e. $\mathrm{lca}_T(x, y) = \rho$. Consequently, (x, y) is an arc in $\vec{G}(T, \sigma)$ for all $y \in V(\vec{G})$ with $\sigma(y) \neq \sigma(x)$. By construction of S in Algorithm 1, we have

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

Algorithm 1 recurses on each such connected component $(\vec{G}_v, \sigma_{|.})$ using a newly created vertex $v \in \mathrm{child}_T(\rho)$ to initialize the tree $T(v)$. By Lemma 3, it clearly holds that, for any $x, y \in L(T(v)) = V(\vec{G}_v)$, (x, y) is an arc in $\vec{G}(T, \sigma)$ if and only it is an arc in $\vec{G}(T(v), \sigma)$. Thus, it suffices to consider only the subtree $T(v)$. Now, we can apply the same arguments as in the previous recursion step to conclude that all arcs incident to the support set $S_2^{(2)}$ constructed in the current recursion step are the same in $(\vec{G}, \sigma)$ and $\vec{G}(T, \sigma)$ and that neither $(\vec{G}, \sigma)$ nor $\vec{G}(T, \sigma)$ contains arcs between distinct connected components of $(\vec{G}_v - S_2^{(2)}, \sigma_{|.})$. Hence, it suffices to consider the connected components of $(\vec{G}_v - S_2^{(2)}, \sigma_{|.})$. Repeated application of this argumentation results in a chain of connected components that are contained in each other. Since Algorithm 1 returns a tree, this chain is finite, say with last element $(\vec{G}_w - S_k^{(2)}, \sigma_{|.})$, and thus $S_k^{(2)} = V(\vec{G}_w)$. In particular, every vertex in $V(\vec{G})$ is contained in the support set of some recursion step. Therefore, all arcs of $\vec{G}(T(v), \sigma_{|.})$ are arcs of $\vec{G}$.

In summary, we have shown that $\vec{G}(T, \sigma) = (\vec{G}, \sigma)$. Hence, $(\vec{G}, \sigma)$ is a connected 2-BMG and, by Lemma 10, T is the unique least resolved tree of $(\vec{G}, \sigma)$. □

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

Lemma 11 *Let $(\vec{G}, \sigma)$ be a 2-BMG and S_ρ the support leaves of the root ρ of its LRT (T, σ). If $W := U(\vec{G}, \sigma) \setminus S_\rho \neq \emptyset$, then the following statements hold:*

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

Proof: From Theorem 3 and the definition of the support set S of $(\vec{G},\sigma)$, we have $S_\rho = S \subseteq U(\vec{G},\sigma)$, and thus $U(\vec{G},\sigma) = S_\rho \uplus W$. Moreover, by Lemma 7, the connected components of $(\vec{G}-S_\rho,\sigma_{|.})$ are exactly the BMGs $\vec{G}(T(v),\sigma_{|.})$ with $v \in \mathrm{child}(\rho)\setminus S_\rho$. Also, recall that the vertices $v \in \mathrm{child}(\rho)\setminus S_\rho$ are all inner vertices of T since the support leaves S_ρ are exactly the children of ρ that are leaves. This together with the contraposition of Lemma 5 implies that $T(v)$ contains both colors.

(1): Let $x \in W$. Since $x \notin S_\rho$, x is contained in some connected component of $(\vec{G}-S_\rho,\sigma_{|.})$, say $\vec{G}(T(v),\sigma_{|.})$ for some $v \in \mathrm{child}_T(\rho)\setminus S_\rho$. Now assume, for contradiction, that $\vec{G}-S_\rho$ has more than one connected component. By Lemmas 5 and 7, there is a vertex $v' \in \mathrm{child}_T(\rho)\setminus S_\rho$ such that $v \neq v'$ and both subtrees $T(v)$ and $T(v')$ contain both colors. Hence, there are distinct $y \in L(T(v))$ and $y' \in L(T(v'))$ with $\sigma(y)=\sigma(y') \neq \sigma(x)$. This together with $x \in L(T(v))$ yields $\mathrm{lca}_T(x,y) \preceq_T v \prec_T \rho = \mathrm{lca}_T(x,y')$, which implies $(x,y') \notin E(\vec{G})$. However, $x \in W \subseteq U(\vec{G},\sigma)$ and $\sigma(y') \neq \sigma(x)$ imply $(x,y') \in E(\vec{G})$; a contradiction. Hence, we conclude that $\vec{G}-S_\rho$ has exactly one connected component, and thus ρ has a single inner vertex child v. Since T is phylogenetic, the latter implies that ρ must be incident to at least one leaf, i.e. $S_\rho \neq \emptyset$. This together with Theorem 3 implies that $\vec{G}$ is connected. In summary, (1) holds.

(2): Let $x \in W$ as in the proof of (1). Using the same arguments as in the proof of (1), we conclude that $\sigma(x)=\sigma(y)$ for every $y \in S_\rho$, otherwise we would obtain $(x,y) \notin E(\vec{G})$, contradicting $x \in U(\vec{G},\sigma)$. Since $x \in W$ was chosen arbitrarily and S_ρ is non-empty, we immediately obtain that all vertices in $U(\vec{G},\sigma) = S_\rho \uplus W$ have the same color, and (2) holds.

(3): Now consider the single inner vertex child v of ρ (cf. (1)), and its set of support leaves S_v, which must be non-empty by Lemma 6. Note that W must be entirely contained in $L(T(v))$ and recall that all vertices in $S_\rho \uplus W$ are of the same color (cf. (2)). First suppose, for contradiction, that S_v only contains vertices of *opposite* color to the vertices in $S_\rho \uplus W$. This immediately implies $S_v \cap W = \emptyset$, thus every vertex $x \in W$ must be located in a subtree $T(w)$ of some inner vertex child w of v. Again by contraposition of Lemma 5, every such $T(w)$ contains both colors. However, this contradicts $(x,y) \in E(\vec{G})$ for every $y \in S_v$, which must hold as a consequence of $x \in W \subset U(\vec{G},\sigma)$ and $\sigma(y) \neq \sigma(x)$. Next suppose, for contradiction, that S_v only contains vertices of the *same* color as the vertices in $S_\rho \uplus W$. In this case, we obtain that the edge ρv is redundant w.r.t. $(\vec{G},\sigma)$. To see this, consider an arc $(x,y) \in E(\vec{G})$ such that $\mathrm{lca}_T(x,y)=v$. Clearly, x must be directly incident to v, since otherwise the subtree below v to which x belongs would contain both colors by Lemma 5, contradicting $(x,y) \in E(\vec{G})$. In other words, every such vertex x is a support leaf of v, thus $\sigma(x) \in \sigma(S_v) = \sigma(S_\rho)$ and $\sigma(y) \notin \sigma(S_\rho)$. In particular, there exists no arc $(x,y) \in E(\vec{G})$ such that $\mathrm{lca}_T(x,y)=v$ and $\sigma(y) \in \sigma(L(T)\setminus L(T(v))) = \sigma(S_\rho)$ and therefore, by Lemma 1, the inner edge ρv is redundant. However, this contradicts the fact that T is least resolved. In summary, only the case in which $S_v \neq \emptyset$ contains vertices of both colors is possible, and thus (3) holds.

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

We now use Lemma 11 to investigate the complexity of Algorithm 1.

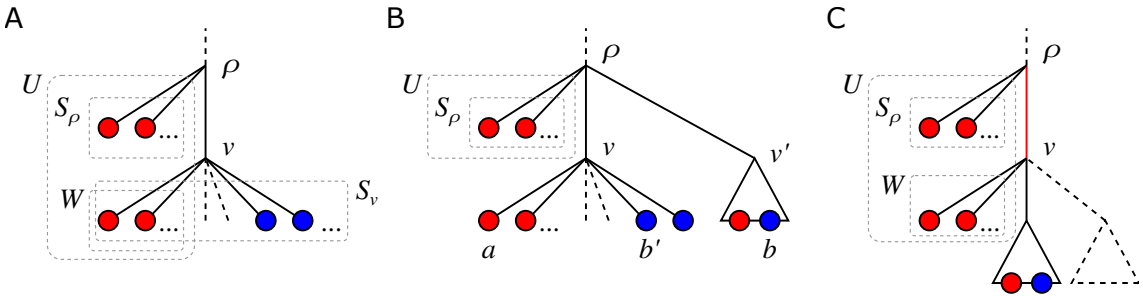

Figure 3: Illustration of Lemma 11. (A) The (local) situation if $W = U \setminus S_\rho \neq \emptyset$ as implied by Lemma 11. In particular, ρ has a single inner vertex child v, all vertices in $U = S_\rho \uplus W$ have the same color, S_v contains vertices of both colors, and $W \subsetneq S_v$. (B) There cannot be a second inner vertex child v', since then none of the vertices except those in S_ρ can be umbrella vertices, e.g. (a, b) is not an arc in the digraph explained by the tree in (B). Hence, this situation is not possible for $W \neq \emptyset$. (C) If S_v does not contain vertices of both colors, then the edge ρv is redundant in the tree, contradicting that (T, σ) in Lemma 11 is the LRT.

Lemma 12 *Algorithm 1 can be implemented to run in $O(|E| \log^2 |V|)$ time for a connected properly 2-colored digraph.*

Proof: Since $\vec{G}$ is connected, we have $|E| \geq |V| - 1$, and thus $|V| \in O(|E|)$. We start from a properly 2-colored digraph represented by its adjacency list and a coloring map. If not stated otherwise, we use hash-based data structures for all sets, maps, and the adjacency list. This enables constant-time lookup and update. To construct maps $\mathrm{outdeg}(x)$, $N^+(x)$, and $N^-(x)$ for the out-degrees, lists of out-neighbors, and lists of in-neighbors, of all vertices $x \in V$, respectively, we traverse the edges in the adjacency list and update the maps accordingly. Since all updates require constant time, this is achieved with $O(|E|)$ operations. We count the number of vertices of each of the two colors. Since $|V| \in O(|E|)$, this also requires $O(|E|)$ operations using constant-time lookups. The initial umbrella set U is then obtained by listing the vertices with maximal out-degree in the color class and checking whether the out-degree equals the number of vertices of the opposite color. We show below that the total effort of constructing *all* sets $S^{(1)}$ and $S^{(2)}$ is $O(|E|)$. The initial umbrella set U and the sets $S^{(1)}$ and $S^{(2)}$ thus can be constructed in linear time. In each recursive call of `Build2ColLRT`, at least one leaf is split off, hence the recursion depth is $|V| - 1$ in the worst case. Since the support vertices removed in each step have all of their in-neighbors in U, their removal does not affect the out-neighborhood for any $x \in V(\vec{G} - U) \subseteq V(\vec{G} - S^{(2)})$, and hence, $\mathrm{outdeg}(x)$ does not require updates. The in-neighborhoods $N^-(x)$ can be updated by removing all out-arcs of the elements in $S^{(2)}$. Since every arc appears exactly once in a removal operation, the total effort for these updates is $O(|E|)$.

Let us assume for the moment that the umbrella vertices U can be obtained efficiently in each step. We shall see below that this is indeed the case. We continue by showing that each vertex needs to be considered at most twice as an umbrella vertex, and that the total effort of constructing all sets $S^{(1)}$ and $S^{(2)}$ is $O(|E|)$. To this end, we distinguish, for each of the single recursion steps, two cases: $S^{(1)} = U$ and $S^{(1)} \subsetneq U$. First if $S^{(1)} = U$, and thus also $S^{(2)} = S^{(1)} = U$, we consider each in-arc of $x \in U$. Since these vertices and their corresponding arcs are removed when constructing $\vec{G} - S^{(2)}$, they are not considered again in a deeper recursion step. In the second case, we have

$S^{(1)} \subsetneq U$, which together with $S^{(2)} = S^{(1)}$ implies $W := U \setminus S^{(2)} \neq \emptyset$, and only the vertices in $U \setminus W$ are removed. However, Lemma 9 and Lemma 11(4) guarantee that, for a 2-BMG as input digraph, the vertices in W appear as support leaves in the next step and thus in the update of U, $S^{(1)}$, and $S^{(2)}$ no more than a second time. In order to use the properties in Lemma 11 for the general case (i.e. $(\vec{G}, \sigma)$ is not necessarily a BMG), we can, whenever $W \neq \emptyset$, (i) check that $\vec{G} - S^{(2)}$ only has a single connected component $\vec{G}_v$, and (ii) pass down the set W to the recursion step on $\vec{G}_v$ in which the condition $W \subsetneq S^{(2)}$ is checked. If any of these checks fails, we can exit false. This way, we ensure that every vertex appears at most twice as an umbrella vertex in the general case. To construct $S^{(1)}$ from U, we have to scan the in-neighborhood $N^-(x)$ of each vertex $x \in U$ and check whether $N^-(x) \subset U$. We repeat this step to construct $S^{(2)}$ from $S^{(1)}$. Membership in U and $S^{(1)}$, respectively, can be checked in constant time (e.g. by marking the vertices in the current set U). Since we have to consider each vertex, and hence, each in-neighborhood at most twice, all sets $S^{(1)}$ and $S^{(2)}$ can be obtained with a total effort of $O(|E|)$.

It remains to show that the input digraph can be decomposed efficiently in such a way that the connectivity information is maintained and the candidates for umbrella vertices in each component are updated. The connected components $\vec{G}_v$ can be obtained by using the dynamic data structure described in [8], often called HDT data structure. It maintains a maximal spanning forest representing the underlying undirected graph with edge set $\widetilde{E} = \{xy \mid (x,y) \in E \text{ or } (y,x) \in E\}$, and allows deletion of all $|\widetilde{E}| \in O(|E|)$ edges with amortized cost $O(\log^2 |V|)$ per edge deletion. The explicit traversal of the connected components to compute U can be avoided as follows: Since $\mathrm{outdeg}(x)$ does not require updates, we can maintain a doubly-linked list of vertices x for each color $i \in \{1,2\}$, and each value of $\mathrm{outdeg}(x)$ where $\sigma(x) = i$. In order to be able to access the highest value of the out-degrees, we maintain these values together with pointer to the respective doubly-linked list in balanced binary search trees (BST), one for each color and each connected component. We will repeatedly make use of the fact that BSTs support searching, inserting, and deleting an item in $O(\log n)$ [9, Section 6.2.3], where n is the total number of items. Since we have at most $|V|$ distinct out-degrees, all three operations require $O(|V|)$ time. The BSTs for the two colors are computed first for $(\vec{G}, \sigma)$ in $O(|V| \log(|V|))$ time and afterwards updated to fit with the out-degree of the component $\vec{G}_v$ currently under consideration. To update these lists and BSTs for $\vec{G}_v$, observe first that $\vec{G}_v$ can be obtained from G by stepwise deletion of single arcs, i.e. edges in the HDT data structure representing the underlying undirected versions. We update, resp., construct the pair of BSTs (one for each color) for each connected component as follows: Since a single arc deletion splits a connected component $\vec{G}'$ into at most two connected components $\vec{G}_1$, and $\vec{G}_2$, we can apply the well-known technique of traversing the smaller component [17]. The size of each connected component can be queried in $O(1)$ time in the HDT data structure. Suppose w.l.o.g. that $|V(\vec{G}_1)| \leq |V(\vec{G}_2)|$. We construct a new pair of BSTs for $\vec{G}_1$, and delete the vertices $V(\vec{G}_1)$ and the respective degrees from the two original BSTs for $\vec{G}$, which then become the BSTs for $\vec{G}_2$. More precisely, we delete each vertex $x \in V(\vec{G}_1)$ in the respective list corresponding to $\mathrm{outdeg}(x)$. Whenever the length of this list drops to zero, we also remove the corresponding out-degree in the BST. Likewise, we insert the out-degree of x and an empty doubly-linked list into the newly-created BST for $\vec{G}_1$, if it is not yet present, and append x to this list. Note that the number of out-degree deletions and insertions does not exceed $|V(\vec{G}_1)|$. Due to the technique of traversing the smaller component, every vertex is deleted and inserted at most $\lfloor \log |V| \rfloor$ times. Therefore, we obtain an overall complexity of $O(|V| \log^2 |V|)$ for the maintenance of the BSTs, where the additional log-factor originates from rebalancing the BSTs whenever necessary.

In each recursion step, the set U can now be obtained by listing (at most) the vertices with

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

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

As a direct consequence of Theorem 3, the LRT of a disconnected digraph $\vec{G}$ is obtained by connecting the roots of the LRTs of the connected components to an additional root vertex, see also [6, Corollary 4].

Theorem 5 *The LRT of a 2-BMG can be computed in* $O(|V| + |E|\log^2|V|)$.

Proof: The connected components $\vec{G}_i = (V_i, E_i)$ of $\vec{G} = (V, E)$ can be enumerated in $O(|V|+|E|)$ operations, e.g. using a breadth-first search on the underlying undirected graph [4, Section 22.2]. By Lemma 12, $O(|E_i|\log^2|V_i|) \leq O(|E_i|\log^2|V|)$ operations are required for each $\vec{G}_i$. Hence, the total effort is $O(|V| + |E| + \log^2|V|\sum_i|E_i|) = O(|V| + |E|\log^2|V|)$. □

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

Both methods for the LRT computation were implemented in Python. Moreover, we note that we did not implement the sophisticated dynamic data structures used in the proof of Lemma 12, but a rather naïve implementation of Algorithm 1. Instead of using the HDT data structure for graph connectivity [8], we compute the (weakly) connected components in Line 10 of each recursion step using the corresponding functions of the Python graph library `NetworkX` [7]. Moreover, we do not employ BSTs to keep track of the out-degrees and candidates for umbrella vertices. Instead, we determine umbrella vertices by iterating through all vertices and checking whether their out-degrees equal the number of vertices of the opposite color. Nevertheless, Figure 4 shows a remarkable improvement of the running time when compared to the general $O(|V|\,|E|\log^2|V|)$ approach for ℓ-BMGs detailed in [6]. Empirically, we observe that the running time of Algorithm 1 indeed scales nearly linearly with the number of edges.

6 Binary-explainable 2-BMGs

Binary phylogenetic trees are of particular interest in practical applications. Not every 2-BMG can be explained by a binary tree. The subclass of *binary-explainable (ℓ-)BMG* are characterized

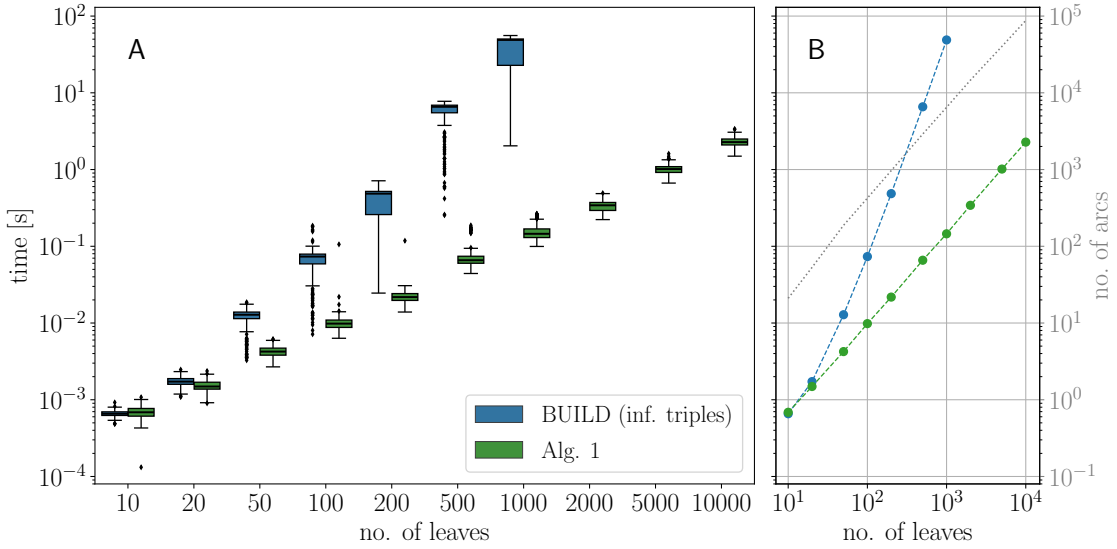

Figure 4: Running time comparison of the general approach for constructing an LRT using `BUILD` (blue) vs. Algorithm 1 (green). For each number of leaves, 200 2-BMGs where generated as described in the text in logarithmic scale. In panel (B), the median are shown on the same time scale as in panel (A). The dotted gray line indicates the median values of the size of the simulated BMGs, i.e. the number of arcs, for comparison (gray scale to the right). LRTs were not computed with the first method for instances with more than 1000 leaves because of the excessive computational cost.

among all BMGs by the absence of single forbidden subgraph called *hourglass* [13, 14], illustrated in Figure 5(A). In this section we briefly describe a modification of Algorithm 1 that allows the efficient recognition of binary-explainable 2-BMGs.

Definition 9 *An* hourglass *in a properly vertex-colored digraph $(\vec{G},\sigma)$, denoted by $[xy \bigtimes\!\!\!\!\!\!\diagup\; x'y']$, is a subgraph $(\vec{G}[Q],\sigma_{|Q})$ induced by a set of four pairwise distinct vertices $Q=\{x,x',y,y'\}\subseteq V(\vec{G})$ such that (i) $\sigma(x)=\sigma(x')\neq\sigma(y)=\sigma(y')$, (ii) $(x,y),(y,x)$ and $(x'y'),(y',x')$ are bidirectional arcs in $\vec{G}$, (iii) $(x,y'),(y,x')\in E(\vec{G})$, and (iv) $(y',x),(x',y)\notin E(\vec{G})$.*

A digraph $(\vec{G},\sigma)$ is *hourglass-free* if it does not contain an hourglass as an induced subgraph. We summarize Lemma 31 and Proposition 8 in [14] in the following proposition.

Proposition 3 *For every BMG $(\vec{G},\sigma)$, the following three statements are equivalent:*

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

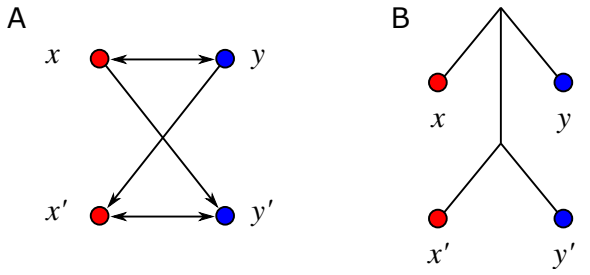

Figure 5: The hourglass in panel (A) is explained by the tree in panel (B).

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

The following Lemma shows that the third condition in Proposition 3 can be translated to a much simpler statement in terms of the support leaves of its LRT.

Lemma 13 *A 2-BMG* $(\vec{G}, \sigma)$ *contains an induced hourglass if and only if its LRT* (T, σ) *contains an inner vertex* u *such that* S_u *contains support vertices of both colors and* $V(\vec{G}(T(u)) - S_u) \neq \emptyset$.

Proof: First, suppose that $(\vec{G}, \sigma)$ contains an hourglass, i.e., by Proposition 3 there is a vertex $u \in V^0(T)$ with distinct children v_1, v_2, and v_3 and two distinct colors r and s satisfying (a) $r \in \sigma(L(T(v_1)))$, $r, s \in \sigma(L(T(v_2)))$, and $s \in \sigma(L(T(v_3)))$, and (b) $s \notin \sigma(L(T(v_1)))$, and $r \notin \sigma(L(T(v_3)))$. Since $(\vec{G}, \sigma)$ is properly 2-colored and (T, σ) is its LRT, Lemma 5 together with $s \notin \sigma(L(T(v_1)))$ and $r \notin \sigma(L(T(v_3)))$ implies that v_1 of color r and v_2 of color s, respectively, are both leaves. In particular, therefore, we know that $v_1, v_2 \in S_u$ are support leaves. By Lemma 7 and since $\vec{G}(T(u), \sigma_{|.})$ is also a BMG, the connected components of $(\vec{G}(T(u)) - S_u, \sigma_{|.}) = (\vec{G}[L(T(u))] - S_u, \sigma_{|.})$ (cf. Lemma 3) are exactly the BMGs $\vec{G}(T(v), \sigma_{|.})$ with $v \in \mathrm{child}(u) \setminus S_u$. This together with the fact that $v_2 \in V^0(T)$ as a consequence of $L(T(v_2))$ containing both colors r and s implies that $(\vec{G}(T(u)) - S_u, \sigma_{|.})$ is not the empty digraph.

Conversely, suppose there is a vertex $u \in V^0(T)$ such that S_u contains support vertices v_1 and v_3 with distinct colors $\sigma(v_1) \neq \sigma(v_3)$ and $V(\vec{G}(T(u)) - S_u) \neq \emptyset$, i.e., u has a child $v_2 \in \mathrm{child}(u) \setminus S_u$ that is not a support leaf and hence satisfies $v_2 \in V^0(T)$. Lemma 5 implies that $L(T(v_2))$ contains both colors since $v_2 \in V^0(T)$. Hence, the three children v_1, v_2, and v_3 of u satisfy conditions (a) and (b) of Proposition 3(3), and thus $(\vec{G}, \sigma)$ contains an induced hourglass. □

Corollary 4 *It can be checked in* $O(|V| + |E| \log^2 |V|)$ *whether or not a properly 2-colored digraph* $(\vec{G}, \sigma)$ *is a binary-explainable BMG.*

Proof: Recall that there is a one-to-one correspondence between the recursion step in Algorithm 1 and the inner vertices $u \in V^0(T)$. As argued in the proof of Lemma 12, every vertex appears at most twice in an umbrella set U. Therefore, it can be checked in $O(|V|)$ total time whether $S = S^{(2)}$ contains vertices of both colors. Since the vertex set of $\vec{G}_u - S_u$ is maintained in the dynamic graph HDT data structure, it can be checked in constant time for each u whether $\vec{G}_u - S_u$ is non-empty. The additional effort to check the condition of Lemma 13 is therefore only $O(|V|)$. Hence, we still require a total effort of $O(|V| + |E| \log^2 |V|)$ (cf. Theorem 5). □

Corollary 4 improves the complexity for the decision whether a 2-BMG is binary-explainable as compared to the $O(|V|^3 \log^2 |V|)$-time algorithm for (general) BMGs presented in [13].

7 Concluding Remarks

We have shown here that 2-BMGs have a recursive structure that is reflected in certain induced subgraphs that correspond to subtrees of the LRT. The leaves connected directly to the root of a given subtree play a special role as support vertices in the corresponding subgraph of the 2-BMG. Since the support vertices of the root can be identified efficiently in a given input digraph, there is a recursive decomposition of $(\vec{G}, \sigma)$ that directly yields the LRT. With the help of a dynamic data structure to maintain connectedness information [8], this provides an $O(|V|+|E| \log^2 |V|)$ algorithm to recognize both 2-BMGs and binary explainable 2-BMGs and to construct the corresponding LRT. This provides a considerable speed-up compared to the previously known $O(|V||E| \log^2 |V|)$ and $O(|V|^3)$ algorithms. Empirically, we observe a substantial speed-up even if simpler data structures are used to implement Algorithm 1. This observation suggests that the depth of the recursion in Algorithm 1 is only logarithmic for typical instances, since the worst case performance of our naïve implementation is $O(|V|(|V| + |E|))$, with the first factor deriving from the depth of the recursion.

Both the theoretical insights and Algorithm 1 itself have potential applications to the analysis of gene families in computational biology. Real-life data necessarily contain noise, and thus likely will deviate from perfect BMGs, naturally leading to graph editing problems for BMGs. Like many combinatorial problems in phylogenetics, these are NP-complete [15] and hence require approximation algorithms and heuristics. The support leaves introduced here provide an avenue to a new class of heuristics, conceptually distinct from approaches that attempt to extract consistent subsets of triples from $\mathcal{R}(\vec{G}, \sigma)$.

References

[1] G. Abrams and J. K. Sklar. The graph menagerie: Abstract algebra and the mad veterinarian. *Math. Mag.*, 83:168–179, 2010. doi:10.4169/002557010X494814.

[2] A. Aho, Y. Sagiv, T. Szymanski, and J. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM J Comput*, 10:405–421, 1981. doi:10.1137/0210030.

[3] H. Cohn, R. Pemantle, and J. G. Propp. Generating a random sink-free orientation in quadratic time. *Electr. J. Comb.*, 9:R10, 2002. doi:10.37236/1627.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, Cambridge, Mass, 2nd edition, 2001.

[5] S. Das, P. Ghosh, S. Ghosh, and S. Sen. Oriented bipartite graphs and the goldbach graph, 2021. arXiv:1611.10259.

[6] 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. *J. Math. Biol.*, 78:2015–2057, 2019. doi:10.1007/s00285-019-01332-9.

[7] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using NetworkX. In G. Varoquaux, T. Vaught, and J. Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11–15, Pasadena, CA USA, 2008.

[8] J. Holm, K. de Lichtenberg, and M. Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *J. ACM*, 48:723–760, 2001. doi:10.1145/502090.502095.

[9] D. E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, Reading, MA, 3rd edition, 1998.

[10] A. Korchmaros. Circles and paths in 2-colored best match graphs, 2020. arXiv:2006.04100.

[11] A. Korchmaros. The structure of 2-colored best match graphs, 2020. arXiv:2009.00447.

[12] D. Schaller, M. Geiß, E. Chávez, M. González Laffitte, A. López Sánchez, B. M. R. Stadler, D. I. Valdivia, M. Hellmuth, M. Hernández Rosales, and P. F. Stadler. Corrigendum to "Best Match Graphs". *J. Math. Biol.*, 82:47, 2021. doi:10.1007/s00285-021-01601-6.

[13] D. Schaller, M. Geiß, M. Hellmuth, and P. F. Stadler. Best match graphs with binary trees. In C. Martín-Vide, M. A. Vega-Rodríguez, and T. Wheeler, editors, *Algorithms for Computational Biology, 8th AlCoB*, volume 12715 of *Lect. Notes Comp. Sci.*, pages 82–93, 2021. doi:10.1007/978-3-030-74432-8_6.

[14] D. Schaller, M. Geiß, P. F. Stadler, and M. Hellmuth. Complete characterization of incorrect orthology assignments in best match graphs. *J. Math. Biol.*, 82:20, 2021. doi:10.1007/s00285-021-01564-8.

[15] D. Schaller, P. F. Stadler, and M. Hellmuth. Complexity of modification problems for best match graphs. *Theor. Comp. Sci.*, 865:63–84, 2021. doi:10.1016/j.tcs.2021.02.037.

[16] C. Semple and M. Steel. *Phylogenetics*. Oxford University Press, Oxford UK, 2003.

[17] Y. Shiloach and S. Even. An on-line edge-deletion problem. *J. ACM*, 28:1–4, 1981. doi:10.1145/322234.322235.