Erschienen in: Discrete Applied Mathematics ; 157 (2009), 10. - S. 2361-2369
https://dx.doi.org/10.1016/j.dam.2008.06.031

Phylogenetic graph models beyond trees

Ulrik Brandes, Sabine Cornelsen*

Department of Computer & Information Science, University of Konstanz, Box D 67, 78457 Konstanz, Germany

Keywords:
Phylogenetic trees
Graph models
Splits
Compatibility
Cactus model

ABSTRACT

A graph model for a set $\mathfrak{S}$ of splits of a set X consists of a graph and a map from X to the vertices of the graph such that the inclusion-minimal cuts of the graph represent $\mathfrak{S}$. Phylogenetic trees are graph models in which the graph is a tree. We show that the model can be generalized to a cactus (i.e. a tree of edges and cycles) without losing computational efficiency. A cactus can represent a quadratic rather than linear number of splits in linear space. We show how to decide in linear time in the size of a succinct representation of $\mathfrak{S}$ whether a set of splits has a cactus model, and if so construct it within the same time bounds. As a byproduct, we show how to construct the subset of all compatible splits and a maximal compatible set of splits in linear time. Note that it is $\mathcal{NP}$-complete to find a compatible subset of maximum size. Finally, we briefly discuss further generalizations of tree models.

1. Introduction

The goal of phylogenetic analysis is to determine and describe the evolutionary relationship between species (taxa). A phylogenetic tree describes in particular the branching process when during time a species is divided into two separate species. One method of obtaining such an evolutionary tree is to consider a bunch of properties (binary characters) that the actual species may or may not have. Then the goal is to construct a tree such that in particular the leaves are labeled with the different species and the properties correspond to the edges: The sets of species mapped to the two connected components of the tree deleting one edge corresponds to the set of species having or not having the corresponding property. Hence, a binary character induces a split, i.e., a partition of the set of taxa into two non-empty parts. In the following, we assume that each split is given by the smaller of its two subsets.

Not every set of splits can be represented in a phylogenetic tree. The splits have to be pairwise compatible, i.e., the intersection of two splits S and T has to be S, T, or the empty set. Given a set $\mathfrak{S}$ of m pairwise compatible splits of a set X of n taxa, Gusfield [19] showed how to construct an evolutionary tree in $\mathcal{O}(mn)$ time. Although he also gives a matching lower bound for the worst case, Agarwala et al. [1] improved the running time for constructing a phylogenetic tree to $\mathcal{O}(n + m + f)$ time, where $f \leq mn/2$ is the sum of the sizes of all splits. The Buneman graph [8] or the tree-popping algorithm of Meacham [27,28] are other approaches for constructing the phylogenetic tree of a set of compatible splits. One way of handling incompatible sets of splits is to compute a significant compatible subset of splits. It was shown by Day and Sankoff [10] that the problem of finding a maximum compatible subset of splits is $\mathcal{NP}$-complete. On the other hand, it is well known, that a maximal compatible subset of splits can be found greedily. We sketch how the greedy algorithm can be implemented to run in $\mathcal{O}(n + m + f)$ time. Both, a maximum and a maximal compatible subset of splits have the disadvantage that they are not unique. Thus, we consider the subset of splits of $\mathfrak{S}$ that are pairwise compatible with all other splits in $\mathfrak{S}$. This subset of all compatible splits is also known as the splits of the loose consensus tree [6] or the kernel splits [3]. A comparison of this set of splits to other compatible subsets of splits can be found in [7, Chapter 6].

* Corresponding author. Tel.: +49 7531 88 4431; fax +49 7531 88 3577.
E-mail addresses: Ulrik.Brandes@uni-konstanz.de (U. Brandes), Sabine.Cornelsen@uni-konstanz.de (S. Cornelsen).

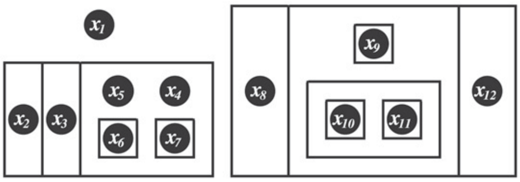


Fig. 1. A drawing of the set of splits in Fig. 2 with axis-parallel rectangles.

Recently, McConnell [25] gave a linear time algorithm for constructing a generalized PQ-tree from which the set of all compatible splits can be deduced. The algorithm is based on the overlap components of Dahlhaus [9]. We give a different algorithm that computes the subset of all compatible splits in $\mathcal{O}(n+m+f)$ time. Both, the algorithm of Dahlhaus [9] and our algorithm use lexicographical sorting as a basic construction step, however, for completely different purposes. We believe that our algorithm is easier to understand and to implement. It also leads to a complete characterization of incompatible splits in terms of prefix trees.

Another way of handling incompatible sets of splits is to extend phylogenetic trees to more complex networks. An overview on the different kinds of phylogenetic networks and their construction can be found, e.g., in [24,23,30]. Two basic types of networks representing splits are: Recombination networks (see e.g. [21]) and networks like splits graphs [2,17] that represent incompatible splits by some minimal cuts of a graph. We use a representation that is similar to the latter case.

A graph model for a set of splits is a graph in which some of the vertices are labeled by the taxa such that there is a one-to-one correspondence between the minimal cuts of the graph and the splits. Note that a phylogenetic tree is a tree model. A cactus is a tree of edges and cycles. Both, a tree model and a cactus model require only $\mathcal{O}(n)$ space. While a tree model can represent linearly many, cactus models can represent up to a quadratic number of splits. A cactus model for the set of splits in Fig. 2(a) is given in Fig. 5. Note that galled trees [21] are also trees of edges and cycles. However, they belong to the category of recombination networks and represent sets of splits differently.

Originally, the cactus model was introduced by Dinitz et al. [13] to represent the set of all minimum cuts of a connected graph. Dinitz and Nutov [14] later characterized all sets of splits that have a cactus model. While the proof is constructive [15], it appears to be difficult to translate it into an algorithm. So far, efficient algorithms are known for constructing the cactus of all minimum cuts of a graph [12,18,31,33]. A cactus model can also be deduced from the generalized PQ-tree of McConnell [25].

Sets of splits that have a cactus model also have a characterization in terms of graph drawing. They are exactly the sets of splits that have a drawing with axis-parallel rectangles [5]. See Fig. 1 for such a drawing. In this paper, we give an algorithm that decides in $\mathcal{O}(n+m+f)$ time whether a set of splits can be represented in a cactus, and if so constructs the model in the same asymptotic running time. The construction is based on the tree model of the subset of all compatible splits. In addition it uses only some easy counting and sorting arguments. In the conclusion, we discuss that our algorithm extends to graph models consisting of trees of edges and cliques or trees of edges, cycles, and cliques, respectively. Sets of splits having such graph models have been characterized in [29,15], respectively.

The paper is organized as follows. In Section 2, we give basic definitions and introduce the necessary concepts. We define the graph model of a set of splits in Section 3. In Section 4 we recall how to construct a tree model for a compatible set of splits utilizing so-called tries. In Sections 5 and 6 we show how to construct a maximal compatible subset and the subset of all compatible splits, respectively. Finally, in Section 7 we examine the existence and construction of a cactus model. We conclude in Section 8.

2. Preliminaries

Throughout this paper, let $X = \{x_1, \dots, x_n\}$ denote a finite set of n taxa. We will denote the *size* n of the set X by $|X|$. By $S \subset X$ we denote that S is a subset of X including that S might be equal to X. A *split* of X is the unordered pair $\{S, X \setminus S\}$ such that $\emptyset \subsetneq S \subsetneq X$. We say that S induces $\{S, X \setminus S\}$. We will assume that the splits are given by the smaller subset. So, throughout this paper, let $\mathfrak{S}$ denote a set of m non-empty subsets S of X such that $|S| \le |X \setminus S|$ and such that $\{S, X \setminus S\} \neq \{T, X \setminus T\}$ for two elements $S, T \in \mathfrak{S}$. We will refer to the elements of $\mathfrak{S}$ also as splits. Further, let $f = \sum_{S \in \mathfrak{S}} |S|$. Fixing an ordering $x_1, \dots, x_n$ of X, a split can be represented by a *characteristic vector*. The characteristic vector of the split induced by S is the vector $(v_1, \dots, v_n) \in \{0, 1\}^n$ such that for all $i = 1, \dots, n$ we have $v_i = 1$ if and only if $x_i \in S$. Hence, a set of splits can be represented by a matrix where the columns are the characteristic vectors of the splits (provided an ordering of the splits is fixed). An example is given in Fig. 2(a). A more succinct way of representing a split $\{S, X \setminus S\}$ is to represent the set S that induces it by a *member list*, i.e., the list of elements of S. An example is given in Fig. 2(b). Throughout this paper we assume that splits are given in this succinct representation.

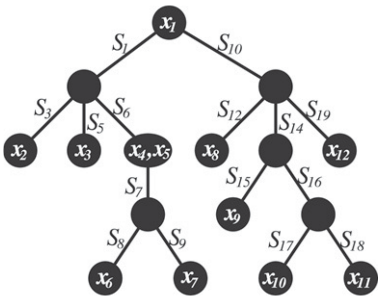
Let $A = (a_{ij})_{i=1,\dots,n,j=1,\dots,m}$ be the matrix of characteristic vectors of the ordered set $\mathfrak{S} = \{S_1, \dots, S_m\}$ of splits with respect to an ordering $x_1, \dots, x_n$ of the taxa. We say that the splits are *lexicographically sorted* with respect to the fixed ordering of the taxa if the columns of A are sorted lexicographically, i.e., if for $1 \le j < k \le m$ it holds that there is an $1 \le \ell \le n$ such that $a_{\ell j} = 1$, $a_{\ell k} = 0$ and $a_{ij} = a_{ik}$, $i < \ell$. Analogously, we say that the taxa are lexicographically sorted with respect to an ordering of the splits if the columns of A are lexicographically sorted. A lexicographical sorting of splits or taxa can be constructed in $\mathcal{O}(n + m + f)$ time using partition refinement [34].

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	S_{18}	S_{19}
x_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x_2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x_3	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x_4	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
x_5	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
x_6	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
x_7	1	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
x_8	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
x_9	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0
x_{10}	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	0	0
x_{11}	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	0	1	0
x_{12}	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1

(a) Characteristic vectors.

$S_1 : x_2, x_3, x_4, x_5, x_6, x_7$
$S_2 : x_2, x_3$
$S_3 : x_2$
$S_4 : x_3, x_4, x_5, x_6, x_7$
$S_5 : x_3$
$S_6 : x_4, x_5, x_6, x_7$
$S_7 : x_6, x_7$
$S_8 : x_6$
$S_9 : x_7$
$S_{10} : x_8, x_9, x_{10}, x_{11}, x_{12}$
$S_{11} : x_8, x_9, x_{10}, x_{11}$
$S_{12} : x_8$
$S_{13} : x_9, x_{10}, x_{11}, x_{12}$
$S_{14} : x_9, x_{10}, x_{11}$
$S_{15} : x_9$
$S_{16} : x_{10}, x_{11}$
$S_{17} : x_{10}$
$S_{18} : x_{11}$
$S_{19} : x_{12}$

(b) Membership lists.



(c) Tree model.

Fig. 2. A set $\mathfrak{S} = \{S_1, \ldots, S_{19}\}$ of splits of the set $X = \{x_1, \ldots, x_{12}\}$ represented as (a) characteristic vectors and (b) as membership lists. (c) A tree model for the set $\mathfrak{S}_\parallel = \mathfrak{S} \setminus \{S_2, S_4, S_{11}, S_{13}\}$ of compatible splits. φ is indicated by vertex labels and edge labels indicate the split that the edge represents.

If X is the set of vertices of a graph, a bipartition of X is also called a cut of the graph. A cut $\{S, V \setminus S\}$ of a connected graph $G = (V, E)$ is *minimal* if and only if both S and $V \setminus S$ induce connected subgraphs of G.

A *rooted tree* (T, r) consists of a connected graph T without cycles and a root r. The *leaves* are the vertices of degree one. Vertex ν is the *predecessor* of a vertex μ (and μ is a *child* of ν and of the edge $\{\nu, \mu\}$) if ν is the next vertex after μ on the unique path from μ to the root. The *subtree rooted at* vertex ν is the subgraph of T induced by the vertices μ for which ν is contained on the path from μ to r. The *level* of a vertex ν is the length of the path from ν to the root r. The *subtree root* of a set E' of edges and vertices of T is the vertex ν on the highest level such that E' is contained in the subtree rooted at ν.

3. Graph models and compatibility

A *graph model* for a set $\mathfrak{S}$ of splits of a set X is a pair (G, φ) that consists of an undirected graph $G = (V, E)$ and a mapping $\varphi : X \to V$ such that $\mathfrak{S}$ is *represented* by the minimal cuts of G, i.e., the set of splits represented by $\mathfrak{S}$ is

$$\{\{\varphi^{-1}(S), \varphi^{-1}(X \setminus S)\}; \{S, X \setminus S\} \text{ is a minimal cut of } G\}.$$

A vertex $\nu \in V$ is called *empty* if $\varphi^{-1}(\nu) = \emptyset$.

Two splits $\{S, X \setminus S\}$ and $\{T, X \setminus T\}$ are *compatible* if at least one of the four *corner sets*

$$S \cap T, \quad S \cap (X \setminus T), \quad (X \setminus S) \cap T \quad \text{and} \quad (X \setminus S) \cap (X \setminus T)$$

is empty. In the example in Fig. 2 there are two pairs of splits that are incompatible: splits S_2 and S_4 and splits S_{11} and S_{13}, respectively.

There is a *tree model* (i.e. a graph model (G, φ) such that G is a tree) for a set $\mathfrak{S}$ of splits if and only if all splits in $\mathfrak{S}$ are pairwise compatible [19]. Let $\mathfrak{S}_\parallel \subset \mathfrak{S}$ be the set of splits that are compatible with each split in $\mathfrak{S}$. We will call the splits in $\mathfrak{S}_\parallel$ the *compatible* splits of $\mathfrak{S}$. An example for the tree model of the subset of all compatible splits is given in Fig. 2(c). Further, we call a set $\mathfrak{S}$ of splits *compatible*, if all splits in $\mathfrak{S}$ are pairwise compatible. Pairs of splits, sets of splits, or splits of a set that are not compatible are called *incompatible*.

x_1 :	
x_2 :	$S_1S_2S_3$
x_3 :	$S_1S_2S_4S_5$
x_4 :	$S_1S_4S_6$
x_5 :	$S_1S_4S_6$
x_6 :	$S_1S_4S_6S_7S_8$
x_7 :	$S_1S_4S_6S_7S_9$
x_8 :	$S_{10}S_{11}S_{12}$
x_9 :	$S_{10}S_{11}S_{13}S_{14}S_{15}$
x_{10}:	$S_{10}S_{11}S_{13}S_{14}S_{16}S_{17}$
x_{11}:	$S_{10}S_{11}S_{13}S_{14}S_{16}S_{18}$
x_{12}:	$S_{10}S_{13}S_{19}$

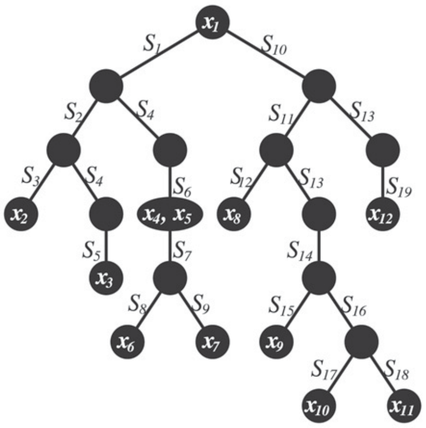


Fig. 3. A trie for the set of splits in Fig. 2.

A *cactus* is a connected graph in which every edge belongs to at most one cycle. A *cactus model* is a graph model (G, φ) such that G is a cactus. Note that the minimal cuts of a cactus are induced by one tree edge or by two edges belonging to the same cycle. A cactus model for a set of splits is given in Fig. 5.

The split induced by $(S \setminus T) \cup (T \setminus S)$ is called the *diagonal split* of two incompatible splits $\{S, X \setminus S\}$ and $\{T, X \setminus T\}$. A set $\mathfrak{S}$ of splits is *crossing* if and only if it holds for every pair of incompatible splits that the splits induced by the four corner sets are in $\mathfrak{S}$ and the diagonal split is not in $\mathfrak{S}$. Dinitz and Nutov [14,15] showed that a set $\mathfrak{S}$ of splits can be represented by a cactus model if and only if it is crossing.

A cactus model (G, φ) is *normal* if there exists no empty vertex v in G with the following two properties: deleting v splits G into exactly two connected components and v is incident to an edge that does not belong to a cycle. Nagamochi and Kameda [32] showed that a normal cactus model is unique up to cycles of length three (which can also be represented as a star) and that any cactus model can be transformed into a normal cactus model in linear time.

4. Tries

Tree models can be constructed via tries [20]. A *trie* (or *prefix tree*) [11] for a set L of strings over an alphabet Σ is a rooted tree (T, r) with the following properties. Each edge of T is labeled with a symbol from Σ. No vertex ν of T has two children μ_1, μ_2 such that $\{\nu, \mu_1\}$ and $\{\nu, \mu_2\}$ have the same label. Let ν be a vertex of T and let $\sigma_1, \ldots, \sigma_k$ be the symbols labeling the edges on the path from r to ν. Then ν represents the string $\sigma_1 \ldots \sigma_k$. The set of strings represented by all leaves is contained in L and the set of strings represented by all vertices of T contains L. A trie can be seen as a deterministic finite automaton. An example for a trie can be found in Fig. 3.

Assume now that $\mathfrak{S} = \{S_1, \ldots, S_m\}$ is lexicographically ordered. Consider for each taxon $x \in X$ a string. It is the list of splits in $\mathfrak{S}$ that contain x, i.e., x is associated with the string $S_{j_1} \ldots S_{j_k}$ such that $j_1 < j_2 < \cdots < j_k$ and $\{S \in \mathfrak{S}; x \in S\} = \{S_{j_1}, \ldots, S_{j_k}\}$. Consider the trie $(T = (V, E), r)$ for the strings associated with the taxa $x_1, \ldots, x_n$. The trie induces a mapping $\varphi : X \to V$. The taxon $x \in X$ is mapped to a vertex ν if the string associated with x equals the sequence of labels on the path from r to ν. See Fig. 3 for an example. If $\mathfrak{S}$ is compatible, then no two edges in T have the same label and (T, φ) is the tree model for $\mathfrak{S}$ [20]. In general, we will refer to (T, φ, r) as the trie constructed for $\mathfrak{S}$.

The trie can be constructed in $\mathcal{O}(n + m + f)$ time [16]. Using a counter for each split, the multiple labels in the trie can be easily determined in $\mathcal{O}(m + f)$ time with any search on the trie. Summarizing, this yields the following theorem stating the same result as in [1].

Theorem 1. *It can be tested in $O(n + m + f)$ time whether a set of splits is compatible and a tree model for a compatible set of splits can be constructed in $\mathcal{O}(n + m + f)$ time.*

5. A maximal compatible subset of splits

A *maximal compatible* subset of $\mathfrak{S}$ is a compatible subset $\mathfrak{S}' \subset \mathfrak{S}$ with the property that $\mathfrak{S}' \cup \{S\}$ is not compatible for any $S \in \mathfrak{S} \setminus \mathfrak{S}'$. We briefly sketch how to compute a maximal compatible subset using a method similar to the one introduced by Gusfield [19]. Let $\mathfrak{S} = \{S_1, \ldots, S_m\}$ be lexicographically sorted with respect to the ordering $x_1, \ldots, x_n$ of the taxa. Since each split is represented by its smaller part, we have the following lemma.

Lemma 2. *Two splits $S_j, S_\ell \in \mathfrak{S}, j < \ell$ are incompatible if and only if $S_j \cap S_\ell \neq \emptyset$ and $S_\ell \setminus S_j \neq \emptyset$.*

π_1	A	B	C	D
x_1	1	1	0	0
x_2	0	0	1	0
x_3	1	0	0	0
x_4	1	1	0	1
x_5	0	0	1	1
x_6	0	0	0	0

π_2	A	B	D	C
x_4	1	1	1	0
x_1	1	1	0	0
x_3	1	0	0	0
x_5	0	0	1	1
x_2	0	0	0	1
x_6	0	0	0	0

π_3	C	D	A	B
x_6	0	0	0	0
x_2	1	0	0	0
x_5	1	1	0	0
x_3	0	0	1	0
x_1	0	0	1	1
x_4	0	1	1	1

Fig. 4. The matrix of the characteristic vectors of four splits with respect to the three lexicographical orderings indicated on page 9. Only D is critical with respect to any of the three indicated orderings of the taxa. However, each of the four splits is high with respect to some ordering.

Proof. Necessity is obvious by definition. So assume that $S_j \cap S_\ell \neq \emptyset$ and $(X \setminus S_j) \cap S_\ell \neq \emptyset$. It remains to show that neither of the other two corner sets is empty. Since $j < \ell$, it follows from the lexicographical sorting that $S_j \not\subset S_\ell$ and hence, $S_j \cap (X \setminus S_\ell) \neq \emptyset$. Finally, $|S_j \cup S_\ell| = |S_j| + |S_\ell| - |S_j \cap S_\ell| \leq n/2 + n/2 - 1 < n$, hence $(X \setminus S_j) \cap (X \setminus S_\ell) \neq \emptyset$. □

Let $S_0 = X$. We define the set $\mathrm{RRL1}(S_\ell) \subset \{S_0, \ldots, S_{\ell-1}\}$ of *recursively rightmost-left-1's* of a split $S_\ell \in \mathfrak{S}$ by including S_j, $0 \leq j < \ell$ in $\mathrm{RRL1}(S_\ell)$ if and only if $|\mathrm{RRL1}(S_j)| \leq 1$ and it exists $1 \leq i \leq n$ such that $x_i \in S_j \cap S_\ell$ and $x_i \notin S_k$ for each $k = j+1, \ldots, \ell-1$ with $|\mathrm{RRL1}(S_k)| = 1$. We call the splits for which the set of recursively rightmost-left-1's contains more than one set the *critical splits*.

The intuition behind the set of recursively rightmost-left-1's is the following. Consider the matrix of characteristic vectors of $\mathfrak{S}$. For each 1 in the ℓth column consider the rightmost column $j < \ell$ such that there is a 1 in the same row and such that S_j is not critical. If there is no such column, let $j = 0$. Then S_j is contained in $\mathrm{RRL1}(S_\ell)$. Consider, e.g., the second matrix in Fig. 4, i.e., the lexicographical ordering A, B, D, C of the splits with respect to the ordering $x_4, x_1, x_3, x_5, x_2, x_6$ of the set X of taxa. $\mathrm{RRL1}(D) = \{B, X\}$ and, hence, D is critical. It follows that $\mathrm{RRL1}(C) = \{X\}$.

Lemma 2 and an argumentation similar to [19] yield the following.

Lemma 3. *Deleting all critical splits from $\mathfrak{S}$ results in a maximal compatible subset of $\mathfrak{S}$.*

Proof. (1) Among two splits that are not compatible at least one is critical: Else, suppose that j_2 is minimal such that there is a $j_1 < j_2$ with the property that the two splits S_{j_1} and S_{j_2} are not compatible and such that neither S_{j_1} nor S_{j_2} is critical. Then, there are $1 \leq i, k \leq n$ with $x_i \in S_{j_1} \cap S_{j_2}$ and $x_k \in S_{j_2} \setminus S_{j_1}$. Let $\mathrm{RRL1}(S_{j_2}) = \{S_j\}$. Then the set S_j is not critical and $x_i, x_k \in S_j$. Hence, by Lemma 2, S_j and S_{j_1} are not compatible. Since j_1 is not critical it follows further that $j_1 < j < j_2$. This contradicts the minimality of j_2.

(2) For any critical split there is a split that is not critical such that the two splits are incompatible: Let $S \in \mathfrak{S}$ be a critical split and let $S_j, S_k \in \mathrm{RRL1}(S)$, $j < k$. Then there are $1 \leq i_1, i_2 \leq n$ with $x_{i_1} \in (S_j \cap S_\ell) \setminus S_k$ and $x_{i_2} \in S_k \cap S_\ell$. Hence, S_k and S are incompatible. S_k is not critical by definition of $\mathrm{RRL1}(S)$. □

Theorem 4. *A maximal compatible subset of splits can be computed in $\mathcal{O}(n + m + f)$ time.*

Proof. We store the recursively rightmost-left-1's of a split S_ℓ in a list L_ℓ. For each taxon, we use a pointer for the currently considered split. At the beginning this pointer points to the first split in the string associated with the taxon.

We do the following step for each $j = 1, \ldots, m$. For each $x \in S_j$, let S be the current split associated with x. (Then $S = S_j$.) Add the predecessor of S in the string w associated with x to L_j. Set the successor of S in w to the currently considered split of x. If in the end of the step for j it holds that L_j contains more than an element, then delete for each $x \in S_j$ the predecessor of the current split from w. □

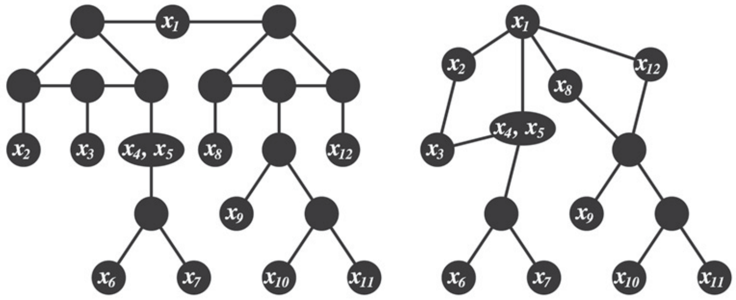
6. The subset of all compatible splits

In this section, we show how splits that are incompatible with some other split can be individuated in linear time. Consider the following lexicographical orderings.

(1) Sort the splits with respect to the ordering $x_1, \ldots, x_n$ of the taxa, resulting in an ordering $S_{\pi_1(1)}, \ldots, S_{\pi_1(m)}$ of $\mathfrak{S}$.
(2) First, sort the taxa with respect to the ordering $S_{\pi_1(1)}, \ldots, S_{\pi_1(m)}$ of the splits, resulting in an ordering $x_{\pi(1)}, \ldots, x_{\pi(n)}$ of X. Then sort the splits with respect to the ordering $x_{\pi(1)}, \ldots, x_{\pi(n)}$ of the taxa, resulting in an ordering $S_{\pi_2(1)}, \ldots, S_{\pi_2(m)}$ of $\mathfrak{S}$.
(3) Sort the splits with respect to the ordering $x_{\pi(n)}, \ldots, x_{\pi(1)}$ of the taxa, resulting in an ordering $S_{\pi_3(1)}, \ldots, S_{\pi_3(m)}$ of $\mathfrak{S}$.

The three lexicographical orderings of the set of splits are illustrated in Fig. 4. We call a split $S_{\pi_i(j_2)}$ *high* with respect to a lexicographical ordering π_i if and only if there is a $j_1 < j_2$ such that $S_{\pi_i(j_2)}$ and $S_{\pi_i(j_1)}$ are not compatible. In Fig. 4, D is high with respect to π_1, D and C are high with respect to π_2, and D, B, and A are high with respect to π_3.

Lemma 5. *Let $S \in \mathfrak{S}$ be a split that is not high with respect to the ordering π_1. Then there are $1 \leq i_1 \leq i_2 \leq n$ such that $S = \{x_{\pi(i)};\ i_1 \leq i \leq i_2\}$.*



(a) Algorithm output.

(b) After normalization.

Fig. 5. Cactus model for splits from Fig. 2.

Proof. Let $S = S_{\pi_1(j_3)}$ and suppose there are i_1, i_2, i_3 with $i_1 < i_2 < i_3$ such that $x_{\pi(i_1)}, x_{\pi(i_3)} \in S$ and $x_{\pi(i_2)} \notin S$. Let $J = \{j \in \{1, \dots, m\};\ j < j_3 \text{ and } x_{\pi(i_1)} \in S_{\pi_1(j)}\}$. Since π is a lexicographical ordering of the taxa, it follows that there are $j_1 \le j_2 < j_3$ with $x_{\pi(i_2)} \in S_{\pi_1(j_2)}$ and $x_{\pi(i_1)} \in S_{\pi_1(j_1)}$. Hence $J \neq \emptyset$. Further, it is not possible that $x_{\pi(i_3)} \in S_{\pi_1(j)}$ for all $j \in J$. Else the row for $x_{\pi(i_2)}$ could not be between the rows for $x_{\pi(i_1)}$ and $x_{\pi(i_3)}$. Hence, there is a $1 \le j < j_3$ with $x_{\pi(i_1)} \in S_{\pi_1(j)}$ and $x_{\pi(i_3)} \notin S_{\pi_1(j)}$. By Lemma 2 it follows that $S_{\pi_1(j_3)}$ and $S_{\pi_1(j)}$ are not compatible. Hence, $S_{\pi_1(j_3)}$ is high. □

Lemma 6. *Let $S \in \mathfrak{S} \setminus \mathfrak{S}_{\|}$. Then there is an $i = 1, \dots, 3$ such that S is high with respect to π_i.*

Proof. Let $S \in \mathfrak{S} \setminus \mathfrak{S}_{\|}$. If S is not high with respect to π_1, then there are $1 \le i_1 \le i_2 \le n$ such that $S = \{x_{\pi(i_1)}, \dots, x_{\pi(i_2)}\}$. If $S = S_{\pi_2(j_1)}$ is not high with respect to π_2, then there is a $j_2 > j_1$ such that S and $S_{\pi_2(j_2)}$ are not compatible. Especially it follows that there is a $k > i_2$ such that $x_{\pi(k)} \in S_{\pi_2(j_2)}$. Hence, $S_{\pi_2(j_2)}$ is before S with respect to π_3 and thus, S is high with respect to π_3. □

It remains to show how to find the high splits. Note that the critical splits are high but the converse is not true. An example is given in Fig. 4. So assume that $\mathfrak{S} = \{S_1, \dots, S_m\}$ is lexicographically sorted with respect to the ordering $x_1, \dots, x_n$ of the taxa. Let (T, φ, r) be the trie constructed in Section 4.

Lemma 7. *The high splits are those that occur more than once as a label in the trie.*

Proof. Let e_1 and e_2 be two edges of T that have the same label S. Let ν be the subtree root of e_1 and e_2. Let μ_i, $i = 1, 2$ be the children of ν such that e_i is contained in the subtree rooted at μ_i. Let S_{j_i}, $i = 1, 2$ be the labels of $\{\nu, \mu_i\}$ and assume without loss of generality that $j_1 < j_2$. Since the indices of the edge labels are strictly increasing on a path from the root to a leaf in T, no edge in the path between ν and e_2 is labeled S_{j_1}. Let $y_1, y_2 \in X$ be such that $\varphi(y_i)$ is contained in the subtree rooted at the child of e_i, $i = 1, 2$. Then $y_1 \in S \cap S_{j_1}$ and $y_2 \in S \cap (X \setminus S_{j_1})$. Hence, by Lemma 2, S and S_{j_1} are not compatible.

Let $S_{j_1}, S_{j_2} \in \mathfrak{S}$, $j_1 < j_2$ be two splits that are not compatible. Then there are $y_1 \in S_{j_1} \cap S_{j_2}$ and $y_2 \in (X \setminus S_{j_1}) \cap S_{j_2}$. Hence, there are edges e, e_1 on the path from r to $\varphi(y_1)$ labeled S_{j_1} and S_{j_2}, respectively, and, since $j_1 < j_2$, it follows that e is before e_1. On the other hand, on the path from r to $\varphi(y_2)$ there is an edge e_2 labeled S_{j_2} but no edge labeled S_{j_1}. Hence $e_1 \neq e_2$. □

Corollary 8. *The subset of all compatible splits can be constructed in $\mathcal{O}(n + m + f)$ time.*

Finally, the proof of Lemma 7 yields an alternative characterization of incompatible splits. For two edges e_1 and e_2 of the trie T let $E(e_1, e_2)$ be the set of edges on the path in T between e_1 and e_2.

Corollary 9. *A split $S \in \mathfrak{S}$ is not compatible if and only if there are two edges e_1, e_2 in the trie T with the following property. e_1 and e_2 have the same label and S is the label of some edge in $E(e_1, e_2)$.*

Proof. Let $1 \le j_1 < j_2 \le m$. Assume first that S_{j_1} is the label of exactly one edge of T. The proof of Lemma 7 yields that S_{j_1} and S_{j_2} are incompatible if and only if there are two edges e_1, e_2 with label S_{j_2} in the trie such that S_{j_1} is the label of some edge in $E(e_1, e_2)$.

Assume now that a split S is the label of at least two edges e_1 and e_2 of T. Then on the one hand S is a label of the edges $e_1, e_2 \in E(e_1, e_2)$ and on the other hand, by Lemma 7, S is not compatible. □

7. Cactus model

In this section, we will show how to test the existence and construct a cactus model for a set of splits in linear time. The algorithm uses parts of the construction given in [15]. Let $(T_{\|}, \varphi, r)$ be the trie for the subset $\mathfrak{S}_{\|}$ of all compatible splits of $\mathfrak{S}$. For a vertex ν of $T_{\|}$ let $X(\nu) \subset X$ be the set of taxa that are mapped to the subtree of $(T_{\|}, r)$ rooted at ν.

For a better understanding of the relation between the tree model of the subset of all compatible splits and the cactus model, suppose first that $\mathfrak{S}$ has a cactus model (G, φ). Note that any pair of incompatible splits must be represented in the same cycle of G. More precisely, let $c : v_1, \ldots, v_\ell$ be a cycle of G. Consider the connected components of the graph $G - c$ that results from G by deleting all edges of c. Let the *base sets* of c be the sets X_i, $i = 1, \ldots, \ell$ of taxa that are mapped to the connected component of $G - c$ containing v_i. Then the cycle c represents the splits $\{X_i, X \setminus X_i\}$, $i = 1, \ldots, \ell$ in $\mathfrak{S}_\|$ and the splits $\{\bigcup_{i=j}^{k} X_i, X \setminus \bigcup_{i=j}^{k} X_i\}$, $1 \leq j < k \leq \ell$, $k - j < \ell - 1$ in $\mathfrak{S}_\| \setminus \mathfrak{S}$.

For example, let c be the cycle on the left-hand side of the cactus model in Fig. 5b. Then $\ell = 4$ and the base sets are $X_1 = \{x_1, x_8, x_9, x_{10}, x_{11}, x_{12}\}$, $X_2 = \{x_2\}$, $X_3 = \{x_3\}$, $X_4 = \{x_4, x_5, x_6, x_7\}$. Cycle c represents 6 splits: the four compatible splits $X \setminus X_1, X_2, X_3, X_4 \in \mathfrak{S}_\|$ and the two incompatible splits $X_3 \cup X_4, X_2 \cup X_3 \in \mathfrak{S} \setminus \mathfrak{S}_\|$.

The cycles in the cactus model correspond to some stars in the tree model of all compatible splits as follows. Note first that for each base set S of a cycle c either $S \in \mathfrak{S}_\|$ or $X \setminus S \in \mathfrak{S}_\|$. Hence, for each base set S there is a vertex ν of $T_\|$ such that $X(\nu) = S$ or $X(\nu) = X \setminus S$. Let $X_1, \ldots, X_\ell$ be the base sets of a cycle c. We distinguish two cases. If there is an $i = 1, \ldots, \ell$ (say $i = \ell$) such that $X \setminus X_i \in \mathfrak{S}$, let ν be the vertex of $T_\|$ such that $X \setminus X_i = X(\nu)$. Since $X \setminus X_i = X_1 \cup \cdots \cup X_{\ell-1}$ it follows that $\{X_i;\ i = 1, \ldots, \ell - 1\} = \{X(\mu);\ \mu \text{ child of } \nu\}$. If $X_i \in \mathfrak{S}$ for all $i = 1, \ldots, \ell$, then $\{X_i;\ i = 1, \ldots, \ell\} = \{X(\mu);\ \mu \text{ child of the root } r\}$. In either case, ν or r, respectively, has to be an empty vertex. Further, the ordering on the base sets induces an ordering on the children of ν or r, respectively, and the $\binom{\ell}{2} - \ell = \ell(\ell-3)/2$incompatible splits represented by c are induced by the unions of the sets represented by at least two and at most $\ell - 1$ consecutive children of ν or r, respectively. Hence, starting with the tree $(T_\|, r)$ we can construct the cycles in G in the following way. First, we assign each split $S \in \mathfrak{S} \setminus \mathfrak{S}_\|$ to the vertex ν of the highest level such that $S \subset X(\nu)$. A vertex that is assigned at least one split will be called a *cycle-replacement* vertex. Then we test for each cycle-replacement vertex whether its children can be ordered such that the assigned splits are unions of the sets represented by consecutive children. Finally, we check whether $\mathfrak{S}$ contains all necessary splits.

We first assign each split in $\mathfrak{S} \setminus \mathfrak{S}_\|$ to exactly one vertex of $T_\|$. Let M_ν be the set of children of a vertex ν of $T_\|$. We call

$$\mathcal{N}(\nu) = \left\{ M \subset M_\nu;\ \bigcup_{\mu \in M} X(\mu) \in \mathfrak{S} \setminus \mathfrak{S}_\| \right\}$$

the *neighbor group* of ν. We say that a split S is a *neighbor* of a vertex ν if $S = \bigcup_{\mu \in M} X(\mu)$ for some $M \in \mathcal{N}(\nu)$.

Lemma 10. (1) *Each split $S \in \mathfrak{S} \setminus \mathfrak{S}_\|$ is the neighbor of exactly one vertex: the subtree root of $\{\varphi(x);\ x \in S\}$.*
(2) *All neighbor groups can be computed in $\mathcal{O}(n + m + f)$ time.*

Proof. (1) Let $S \in \mathfrak{S} \setminus \mathfrak{S}_\|$. Obviously, S cannot be the neighbor of a vertex other than the subtree root of $\{\varphi(x);\ x \in S\}$. Suppose S is not the neighbor of a vertex. Let ν be a vertex of $T_\|$ on the highest level such that $X(\nu) \cap S \neq \emptyset$, and $X(\nu) \cap (X \setminus S) \neq \emptyset$. Then for each child μ of ν either $X(\mu) \subset S$ or $X(\mu) \subset X \setminus S$. Since S is not a neighbor of ν it follows that $S \not\subset X(\nu)$. Hence, $S \cap X(\nu)$, $(X \setminus S) \cap X(\nu)$ and $S \cap (X \setminus X(\nu))$ are not empty. Hence, since $|S|, |X(\nu)| < |X|/2$ it follows also $(X \setminus S) \cap (X \setminus X(\nu)) \neq \emptyset$. This means that S and $X(\nu) \in \mathfrak{S}_\|$ are not compatible, a contradiction.

(2) The vertex ν to which a set $S \in \mathfrak{S} \setminus \mathfrak{S}_\|$ is assigned can be found in $\mathcal{O}(|S|)$ time. We proceed from the vertices $\varphi(x)$, $x \in S$ upwards. Let ν be the predecessor of the currently considered vertex. If ν is found for the first time, add ν to the set R of potential subtree roots. Use a counter to store the number of children of ν and the number of vertices in $\varphi^{-1}(\nu)$ that have already been visited by the algorithm. If all sets represented by all the children of a vertex turn out to be contained in S and $\varphi^{-1}(\nu) \subset S$, then we delete ν from R and continue to proceed upwards. In the end, there is only one vertex ν left in R. Let ℓ be the level of ν and let $\mu_1, \ldots, \mu_k$ be those among the traversed vertices that have level $\ell + 1$. Then $S = V(\mu_1) \cup \cdots \cup V(\mu_k)$. The size of the subtree of $T_\|$ that has to be traversed is linear in the size of S. □

Let ν be a cycle-replacement vertex and let $\mu_1, \ldots, \mu_\ell$ be the children of ν. Let $S \in \mathcal{N}(\nu)$ and let π be a permutation of $\{1, \ldots, \ell\}$. The set S is an *interval* with respect to π if there are $1 \leq i \leq j \leq \ell$ such that $S = \{\mu_{\pi(i)}, \ldots, \mu_{\pi(j)}\}$. The permutation π is a *consecutive-ones ordering* for $\mathcal{N}(\nu)$ if each set in $\mathcal{N}(\nu)$ is an interval with respect to π. If each set in $\mathcal{N}(\nu)$ or its complement is an interval, then π is a *circular consecutive-ones ordering*. The neighbor group $\mathcal{N}(\nu)$ has the *(circular) consecutive-ones property* if there exists a (circular) consecutive-ones ordering for $\mathcal{N}(\nu)$. Our observations in the beginning of Section 7 can now be reformulated as follows.

Lemma 11. *A set $\mathfrak{S}$ of splits has a cactus model if and only if for each cycle-replacement vertex ν in the tree model of $\mathfrak{S}_\|$ it holds that*

(1) *$\mathcal{N}(\nu)$ has the consecutive-ones property if ν is not the root,*
(2) *$\mathcal{N}(\nu)$ has the circular consecutive-ones property if ν is the root,*
(3) *$\mathcal{N}(\nu)$ has degree(ν)(degree$(\nu) - 3)/2$ elements, and*
(4) *ν is empty.*

A (circular) consecutive-ones ordering of neighbor groups can be constructed, if it exists, e.g., using *PQ*-trees [4] or lexicographical breadth-first search [22]. We here give a much simpler algorithm based on the special structure of neighbor groups in the case of crossing splits. Let ν be a cycle-replacement vertex of $T_\|$. We first assume that ν is not the root. Let $\textsc{two}(\nu) \subset \mathcal{N}(\nu)$ be the subset of elements of size two. If there is a cactus model for $\mathfrak{S}$, then there exists an ordering $\pi : \mu_1, \ldots, \mu_\ell$ of the set M_ν of children of ν such that $\textsc{two}(\nu) = \{\{\mu_1, \mu_2\}, \ldots, \{\mu_{\ell-1}, \mu_\ell\}\}$. If $\mathfrak{S}$ is a crossing set of splits, we can find π as follows:

- Let μ_1 be a child of ν that occurs only in one element of $\textsc{two}(\nu)$, say $\{\mu_1, \mu_2\}$. Remove $\{\mu_1, \mu_2\}$ from $\textsc{two}(\nu)$ and set $i = 2$.
- While $i < \ell$ let $\{\mu_i, \mu_{i+1}\} \in \textsc{two}(\nu)$ be the only neighbor containing μ_i. Remove $\{\mu_i, \mu_{i+1}\}$ from $\textsc{two}(\nu)$ and increase i by one.

If $\textsc{two}(\nu)$ does not contain the required sets, $\mathfrak{S}$ does not have a cactus representation. Having found the ordering π, it remains to test whether all elements of $\mathcal{N}(\nu)$ are intervals with respect to π. The algorithm can be implemented straight forwardly in $\mathcal{O}(|\mathcal{N}(\nu)| \cdot \text{degree}(\nu))$ time. Note that in case $\mathfrak{S}$ has a cactus representation $\theta(|\mathcal{N}(\nu)| \cdot \text{degree}(\nu)) = \theta(\text{degree}(\nu)^3) = \theta(\sum_{S \in \mathcal{N}(\nu)} |S|)$.

If the root r is a cycle-replacement vertex, we apply an analogous procedure to the set $\{S;\ S \in \mathcal{N}(r) \text{ or } M_r \setminus S \in \mathcal{N}(r)\}$. Having then found the ordering π, we have to check whether S or its complement is an interval with respect to π. Hence, the procedure finds (circular) consecutive-ones orderings for all neighbor groups in $\mathcal{O}(f)$ time.

Having identified the cycle-replacement vertices and the ordering of their children, and knowing that the conditions for the existence of a cactus model are fulfilled, the final construction is as follows. For the cactus model (G, φ), we start with a copy of $T_\|$.

For each cycle-replacement vertex ν let $\mu_1, \ldots, \mu_\ell$ be the children of ν ordered according to a (circular) consecutive-ones ordering of $\mathcal{N}(\nu)$. Let $\mu'_1, \ldots, \mu'_\ell$ be new vertices. Replace ν by the cycle $\{\nu, \mu'_1\}, \{\mu'_1, \mu'_2\}, \ldots, \{\mu'_{\ell-1}, \mu'_\ell\}, \{\mu'_\ell, \nu\}$ (or by the cycle $\{\mu'_1, \mu'_2\}, \ldots, \{\mu'_{\ell-1}, \mu'_\ell\}, \{\mu'_\ell, \mu'_1\}$, respectively, if ν is the root) and replace each edge $\{\nu, \mu_i\}$ by the edge $\{\mu'_i, \mu_i\}$.

The pseudo-code for the cactus construction can be found in Algorithm 1. The result is a cactus model (G, φ) for $\mathfrak{S}$ which, however, need not be normal. See Fig. 5. It can be normalized in linear time [32] if necessary. Summarizing, we have the following theorem.

Theorem 12. *It can be tested in $\mathcal{O}(n + m + f)$ time whether a set of splits is crossing and a cactus model for a crossing set of splits can be constructed in $\mathcal{O}(n + m + f)$ time.*

Algorithm 1 (*Construction of a Cactus Model*).

```
Input: set $\mathfrak{S}$ of splits of $X = \{x_1, \ldots, x_n\}$
Output: cactus model $(G, \varphi)$ for $\mathfrak{S}$ or information that $\mathfrak{S}$ is not crossing
begin
  Construct the set $\mathfrak{S}_\|$ of all compatible splits of $\mathfrak{S}$ (Section 6);
  Construct the trie $(T, \varphi, r)$ for $\mathfrak{S}_\|$ (Section 4);
  for each inner vertex $\nu$ of $T$ do
    Construct the neighbor group $\mathcal{N}(\nu)$ (Lemma 10);
  Let $(G, \varphi)$ be a copy of $(T, \varphi)$;
  for each inner vertex $\nu$ of $T$ with $\mathcal{N}(\nu) \neq \emptyset$ do
    if Conditions 1–4 of Lemma 11 are fulfilled
      Order the set $M_\nu$ of children of $\nu$ according to
      a (circular) consecutive-ones ordering for $\mathcal{N}(\nu)$;
      Replace $\nu$ in $G$ by a cycle of length degree$(\nu)$;
    else
      $\mathfrak{S}$ is not crossing (i.e. no cactus model exists);
      exit;
end
```

8. Conclusion and Extensions

We presented algorithms for constructing the tree model of a maximal compatible subset of splits and of the subset of all compatible splits, and for testing the existence of and, if it exists, constructing a cactus model. All these are optimal in the sense that they run in time linear in the size of a succinct representation of the splits.

We assumed that the splits are distinct and given by their smaller parts. This is without loss of generality since we can transform an arbitrary (multi-)set $\mathfrak{S}$ of m non-empty proper subsets of a set of n taxa and with $f = \sum_{S \in \mathfrak{S}} |S|$ in $\mathcal{O}(n+m+f)$ time into a set $\mathfrak{S}'$ with the desired properties such that $\mathfrak{S}$ and $\mathfrak{S}'$ induce the same splits. Let $x_1 \in X$ be a fixed taxon. First, for each $S \in \mathfrak{S}$ with $|S| > n/2$ replace S by $X \setminus S$. We break ties, by further replacing each $S \in \mathfrak{S}$ with $|S| = n/2$ and $x_1 \in S$ by $X \setminus S$. Now, after sorting the splits lexicographically, all identical splits are in a row. Crossing sets of splits are *circular* [2], i.e., they have the circular consecutive-ones property. Suppose now that $\mathfrak{S}$ is any circular set of splits. Then Conditions $1 + 2$

of Lemma 11 are fulfilled for all cycle-replacement vertices. Hence, a cactus model for a superset of $\mathfrak{S}$ with the property that the splits in $\mathfrak{S}_{\|}$ are represented by tree edges can be constructed in linear time: For each cycle-replacement vertex ν that is not empty add the set $\varphi^{-1}(\nu)$ to $\mathfrak{S}$. (Eventually divide $\varphi^{-1}(\nu)$ into two sets if ν is the root.) Finally, add the missing sets to $\mathfrak{S}$ to fulfill Condition 3 of Lemma 11.

Crossing sets of splits are especially *semi-crossing* [15], i.e., for any pair of incompatible splits, the splits induced by the four corner sets are in the set. Another specialization of semi-crossing sets applied, e.g., in modular decomposition are the so-called decomposable sets [26,29]. A semi-crossing set of splits is *decomposable* if for any pair of incompatible splits also the diagonal split is in the set. A graph model for a semi-crossing set of splits can be obtained by replacing each star consisting of a cycle-replacement vertex and its children by either a cycle or a complete graph [15]. A graph model for a decomposable set of splits can be obtained by replacing each star consisting of a cycle-replacement vertex and its children by a complete graph [29]. Hence, applying our algorithm, it can be decided in linear time whether a set of splits is semi-crossing or decomposable and if so a graph model for a semi-crossing or decomposable set of splits can also be constructed in linear time.

References

[1] R. Agarwala, D. Fernandez-Baca, G. Slutzki, Fast algorithms for inferring evolutionary trees, Journal of Computational Biology 2 (3) (1995) 397–407.

[2] H.-J. Bandelt, A.W.M. Dress, A canonical decomposition theory for metrics on a finite set, Advances in Mathematics 92 (1992) 47–105.

[3] C. Bonnard, V. Berry, N. Lartillot, Multipolar consensus for phylogenetic trees, Systematic Biology.

[4] K.S. Booth, G.S. Lueker, Testing for the consecutives ones property, interval graphs, and graph planarity using PQ-tree algorithms, Journal of Computer and System Sciences 13 (1976) 335–379.

[5] U. Brandes, S. Cornelsen, D. Wagner, Characterizing families of cuts that can be represented by axis-parallel rectangles, Journal on Graph Algorithms and Applications 9 (1) (2005) 99–115.

[6] K. Bremer, Combinable component consensus, Cladistics 6 (1990) 369–372.

[7] D. Bryant, Building trees, hunting for trees, and comparing trees, Ph.D. Thesis, University of Canterbury, 1997.

[8] P. Buneman, The recovery of trees from measures of dissimilarity, in: F.R. Hodson, D.G. Kendall, P. Tautu (Eds.), Proceedings of the Anglo-Romanian Conference on Mathematics in the Archaeological and Historical Sciences, The Royal Society of London and The Academy of the Socialist Republic of Romania, Edinburgh University Press, 1971, pp. 387–395.

[9] E. Dahlhaus, Parallel algorithms for hierarchical clustering and applications to split decomposition and parity graph recognition, Journal of Algorithms 36 (2000) 205–240.

[10] W.H.E. Day, D. Sankoff, Computational complexity of inferring phylogenies by compatibility, Discrete Applied Mathematics 35 (2) (1986) 224–229.

[11] R. de la Briandais, File searching using variable length keys, in: Proceedings of the Western Joint Computer Conference, 1959, pp. 295–298.

[12] A. De Vitis, The cactus representation of all minimum cuts in a weighted graph, Tech. Rep. 454, Consiglio nazionale delle ricerche, istituto di analisi dei sistemi ed informatica, Roma, Viale Manzoni 30, 00185 Roma, May 1997.

[13] Y. Dinitz, A.V. Karzanov, M. Lomonosov, On the structure of a family of minimal weighted cuts in a graph, in: A. Fridman (Ed.), Studies in Discrete Optimization, Nauka, 1976, pp. 290–306 (in Russian).

[14] Y. Dinitz, Z. Nutov, A 2-level cactus model for the system of minimum and minimum +1 edge-cuts in a graph and its incremental maintenance, in: Proceedings of the 27th Annual ACM Symposium on the Theory of Computing, STOC '95, ACM, The Association for Computing Machinery, 1995, pp. 509–518.

[15] Y. Dinitz, Z. Nutov, Cactus tree type models for families of bisections of a set, manuscript.

[16] S. Dori, G.M. Landau, Construction of aho corasick automaton in linear time for integer alphabets, in: A. Apostolico, M. Crochemore, K. Park (Eds.), Proceedings of the 16th Annual Symposium of Combinatorial Pattern Matching, CPM '05, in: Lecture Notes in Computer Science, vol. 3537, Springer, 2005, pp. 168–177.

[17] A.W.M. Dress, D.H. Huson, Constructing splits graphs, IEEE/ACM Transactions on Computational Biology and Bioinformatics 1 (3) (2004) 109–115.

[18] L. Fleischer, Building chain and cactus representations of all minimum cuts from Hao–Orlin in the same asymptotic run time, Journal of Algorithms 33 (1) (1999) 51–72.

[19] D. Gusfield, Efficient algorithms for inferring evolutionary trees, Networks 21 (1991) 19–28.

[20] D. Gusfield, Algorithms on Strings, Trees, and Sequences, Cambridge University Press, 1997.

[21] D. Gusfield, S. Eddhu, C. Langley, Efficient reconstruction of phylogenetic networks with constrained recombination, in: Proceedings of the 2nd IEEE Computer Society Conference on Bioinformatics, CSB'03, 2003, pp. 363–374.

[22] M. Habib, R.M. McConnell, C. Paul, L. Viennot, Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing, Theoretical Computer Science 234 (1–2) (2000) 59–84.

[23] K.T. Huber, V. Moulton, Phylogenetic networks, in: O. Gascuel (Ed.), Mathematics of Evolution and Phylogeny, Oxford University Press, 2005, pp. 178–204.

[24] D.H. Huson, D. Bryant, Application of phylogenetic networks in evolutionary studies, Molecular Biology and Evolution 23 (2) (2006) 254–267.

[25] R.M. McConnell, A certifying algorithm for the consecutive-ones property, in: Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'04, Society for Industrial and Applied Mathematics, 2004, pp. 768–777.

[26] R.M. McConnell, J.P. Spinrad, Modular decomposition and transitive orientation, Discrete Mathematics 201 (1999) 189–241.

[27] C.A. Meacham, A manual method for character compatibility analysis, Taxon 30 (3) (1981) 591–600.

[28] C.A. Meacham, Theoretical and computational considerations of the compatibility of qualitative taxonomic, in: J. Felsenstein (Ed.), Numerical Taxonomy, in: NATO ASI, vol. G1, Springer, 1983, pp. 304–314.

[29] R.H. Möhring, Algorithmic aspects of the substitution decomposition in optimization over relations, set systems and Boolean functions, Annals of Operations Research 4 (1985–6) 195–225.

[30] D.A. Morrison, Networks in phylogenetic analysis: New tools for population biology, International Journal for Parasitology 35 (2005) 567–582.

[31] H. Nagamochi, T. Kameda, Constructing cactus representation for all minimum cuts in an undirected network, Journal of the Operations Research Society of Japan 39 (2) (1996) 135–158.

[32] H. Nagamochi, T. Kameda, Canonical cactus representation for minimum cuts, Japan Journal of Industrial and Applied Mathematics 11 (3) (1994) 343–361.

[33] H. Nagamochi, Y. Nakao, T. Ibaraki, A fast algorithm for cactus representations of minimum cuts, Japan Journal of Industrial and Applied Mathematics 17 (2) (2000) 245–264.

[34] R. Paige, R.E. Tarjan, Three partition refinement algorithms, SIAM Journal on Computing 16 (6) (1987) 973–989.