have yet to emerge. And widely used methods for the computation of rooted phylogenetic networks (Chapter 6). Their theory is still under development rooted phylogenetic networks in systematic biology. Then we turn to clusters and used in numerous publications in systematic biology. The existing algorithms are routinely been worked out in detail over the last 20 years. The existing algorithms are routinely (Chapter 5). The mathematical and computational aspects of such networks have phylogenetic networks. First we study splits and unrooted phylogenetic networks. In the second part of the book we develop the theoretical foundations for phy-
For computing compatible sets of splits, the split decomposition method provides

inference approaches (described in Chapter 2) can be thought of as methods

Theory of a related mathematical hierarchy using the split decomposition algorithm. We also give a brief introduction to

Weakly compatible splits are of interest because they can be efficiently calculated

While they can be represented by split networks that are not-labeled planar,

Figure 2.1 shows the relationships between some of the main concepts introduced

Splits and unrooted phylogenetic networks

the foundation for the theory of split networks was laid down in a seminal paper

basic building blocks for rooted phylogenetic trees and networks (see Chapter 6),

unrooted phylogenetic networks are among the most important tools for analyzing phylogenetic data and a large class of

are incompatible.

possible way to generalize from trees to networks is to consider sets of splits that

Any set of splits that is compatible corresponds to a phylogenetic tree, and so one

weak compatible tree can correspond to a phylogenetic tree defined by the topology of the tree and the splitting

partition of the underlying set 

It is motivated by the simple but crucial observation that every edge in an unrooted phylogenetic tree defines a partition of the underlying set 

in two non-empty and disjoint subsets A and B, shown as a split. The split of
\[ \{ \emptyset \neq \mathcal{A} \cup B \land A \neq \mathcal{A} \cup \emptyset, \mathcal{A} \subset B \subset \mathcal{A} \cup \emptyset \} = \mathcal{A} \setminus S \]

set of splits induced by \( \mathcal{A} \) on \( \mathcal{A} \) and \( \mathcal{A} \cup S \) is a subset of \( \mathcal{A} \). We define the set of splits induced by \( \mathcal{A} \) to be the split set of \( \mathcal{A} \) and \( \mathcal{A} \cup S \).

Let \( S \) be a set of splits on \( \mathcal{A} \). Let \( \mathcal{A} \) be a subset of \( \mathcal{A} \). We define the to denote the other part.

We use \( S(x) \) to denote the split set that contains \( x \) and we use \( S(x) \) to denote the split set that contains \( \mathcal{A} \).

Formally, \( S(x) = (\text{maxMin} \mid |A| \mid |B|) \). A split of size one is called a trivial split. For any \( \mathcal{A} \subset B \subset \mathcal{A} \cup \emptyset \), the minimal cardinality of the two parts \( \mathcal{A} \) and \( B \) is defined as the minimal cardinality of the two parts \( \mathcal{A} \) and \( B \). More formally, \( S = \mathcal{A} \cap B \). We call \( \mathcal{A} \) and \( B \) the two split parts of \( S \). The size of a split \( S \) equals \( |A| + |B| \) on \( \mathcal{A} \). We call \( A \) and \( B \) the two split parts of \( S \). The size of a split \( S \).

A split is a set of splits that contains \( \mathcal{A} \) and \( \mathcal{A} \cup S \). We also use the notation \( \mathcal{A} = B \cup \emptyset \) and \( \emptyset = B \cup \emptyset \). A split is a partition of a set of taxa into two non-empty subsets \( A \) with \( A \cup B = \mathcal{A} \). We also use the notation \( \mathcal{A} = B \cup \emptyset \) and \( \emptyset = B \cup \emptyset \).

Definition 5.2.1 (Split) A split is a set of splits that contains \( \mathcal{A} \) and \( \mathcal{A} \cup S \). The size of a split \( S \).

Spills and clusters are closely related concepts. While clusters group taxa into distinct collections of taxa that are not related, spills are sets of splits that are not necessarily compatible. We will also discuss the corresponding types of spill networks in Part III.

Spills and unrooted phylogenetic networks

Figure 5.1: Overview of the main concepts introduced in this chapter. On the left, we list the different properties that a set of spills can have, in order of decreasing generality, and in the middle, we list the corresponding types of spill networks. On the right, we list the different types of unrooted phylogenetic networks.
The term encoding is justified by the observation that the tree $T$ can be uniquely
represented by $S(T)$.

$S(T) = \{ (e) \mid e \in E(T) \}$

Let $T$ be an unrooted phylogenetic tree on $\chi$. We define the split encoding $S(T)$
concisely in the following way. Assume that $T$ contains at least one edge. Let $e$ be an
edge of $T$. Define

$S(T)_e = (e) \cup S(T)_p \cup S(T)_q$

where $p$ and $q$ are the two endpoints of $e$, and for $p, q \neq e$. If the unrooted phylogenetic
tree $T$ has no edges, then $S(T)$ is empty.

Let $T$ be an unrooted phylogenetic tree on $\chi$. We define the split encoding $S(T)$
concisely in the following way. Assume that $T$ contains at least one edge. Let $e$ be an
edge of $T$. Define

$S(T)_e = (e) \cup S(T)_p \cup S(T)_q$

where $p$ and $q$ are the two endpoints of $e$, and for $p, q \neq e$. If the unrooted phylogenetic
tree $T$ has no edges, then $S(T)$ is empty.
The analogous result for clusters and rooted phylogenetic trees, which is shown in Section 6.4, is that the result holds if the outgroup tree and the result follows from the root node of the tree. The result was first formulated and proved in [39]. One way to see that the result holds is simply to apply the outgroup trick and then the result follows from Theorem 5.3.2.

Let $S$ be a set of splits on $X$ and assume $S$ contains all trivial splits on $X$. There exists a unique rooted phylogenetic tree $T$ that is compatible with $S$ if and only if $S$ is compatible with $T$.

Theorem 5.3.2 (Compatibility Theorem). Let $S$ be a set of splits on $X$ and assume $S$ contains all trivial splits on $X$. There exists a unique rooted phylogenetic tree $T$ that is compatible with $S$ if and only if $S$ is compatible with $T$.

Otherwise, the two splits are called incompatible. A set of splits $S$ is called compatible if all pairs of splits in $S$ are compatible.

Definition 5.3.1 (Compatibility of splits). Two splits $S_1, S_2$ are compatible if, for any $A, B \subseteq V$, $A \cap S_1 \subseteq B$ or $A \cap S_2 \subseteq B$.

Their split $S$ is empty:

If all splits in $S$ are compatible, then the following four possible intersections of $S_1$ and $S_2$ are possible:

1. $A \cap S_1 \subseteq B$ and $A \cap S_2 \subseteq B$,
2. $A \cap S_1 \subseteq B$ and $A \cap S_2 \subseteq B$,
3. $A \cap S_1 \subseteq B$ and $A \cap S_2 \subseteq B$,
4. $A \cap S_1 \subseteq B$ and $A \cap S_2 \subseteq B$.

Suppose we are given an arbitrary set of splits $S$ on $X$. We would like to know whether $S$ can be represented by some rooted phylogenetic tree $T$. That is, does there exist a set of splits $S$ with $S = S(T)$? The answer is given by the concept of compatibility. Does there exist a rooted phylogenetic tree $T$ on $X$ whose split

\[ \{ [p, q] \in T(a, b), q \in \text{split}(a) \} \]

Consider the following set of splits $S$.

Consider the following set of splits $S$:

1. $A \cap S_1 \subseteq B$ and $A \cap S_2 \subseteq B$,
2. $A \cap S_1 \subseteq B$ and $A \cap S_2 \subseteq B$,
3. $A \cap S_1 \subseteq B$ and $A \cap S_2 \subseteq B$,
4. $A \cap S_1 \subseteq B$ and $A \cap S_2 \subseteq B$.

The set $S(T)$ of all splits associated with an unrooted phylogenetic tree $T$ on $X$ can be computed as follows:

Algorithm 5.2.2 (Splits from trees). The set $S(T)$ of all splits associated with an unrooted phylogenetic tree $T$ on $X$ can be computed as follows:

1. Choose a split $T$ and assume that all edges of $T$ are directed away from $p$.
2. For each edge $e$ of $T$, add the split $o(e)$ to $S(T)$.
3. Add the split $o(T)$ to $S(T)$.
4. Return $S(T)$.
For the set of splinters shown in Figure 5.3(a).

Exercise 5.4.2 (Splinters to clusters example) (a) Using c as an example, list all clusters.

Exercise 6.2.2.

In other words, this is compatible if and only if C is compatible (see Definition 3.4.1 (Splinters to clusters) for this assignment preserves incomparability).

Exercises are present in C.

C = S(0). We usually also consider {0} as a cluster to ensure that all initial clusters are present with S to be the split part that does not contain 0, that is, we set C associated with S to be the split part that does not contain 0. Then, for each split S = \{S⁺, S⁻\} in S, we define the cluster an n-gon region o \in \text{\textit{C}}. Then, for each split S = \{S⁺, S⁻\} in S, we define the cluster.

To obtain a set of clusters C from a set of splinters S on X, we must first choose a partition of clusters in clusters and vice versa.

Splinters and clusters

S₄ (Splinters and clusters)

ji and only if the incomparability graph I(C(S)) has no edges.

| \{S⁺, S⁻\} | \{S⁺, S⁻\} |

S⁺ and S⁻ are incomparable.

\text{Definition 5.3.3 (Incomparability Graph).} The incomparability graph I(C(S)) of a graph that is defined as follows (see Figure 5.3(a).

\text{Lemma 5.3.3.} This assumption can be dropped if we use X, rather than X². This assumption can be dropped if we use X, rather than X².
for clusters in Section 6.2.2, that these two problems are NP-complete.

It follows from the NP-completeness of the two analogous problems formulated
the subset of $\mathcal{X} \subseteq \mathcal{X}$ such that the set of splits $S' \subseteq S$. Determine a maximum

**Problem 5.4.5 (Maximize compatible subsets problem)**

maximize the compatible

The second problem is to remove a minimum number of splits from the set of splits $\mathcal{X}$. If $\mathcal{X}$ is compatible, then there are two basic

**Problem 5.4.4 (Maximize compatible problem)**

maximize the compatible

let $\mathcal{S}$ be a set of splits on $\mathcal{X}$. If $\mathcal{X}$ is compatible, then there are two basic

5.4.1 Optimal compatible subsets

are possible on a set $\mathcal{X}$ of $n$ taxa.

Exercise 5.4.3 (Number of splits and clusters)

from $\mathcal{G} \subseteq \mathcal{S}(\mathcal{T})$, since their union and the set of associated splits directly

In the special case that the set of clusters $\mathcal{C}$ is compatible and this corresponds to

\[ \emptyset \cap \mathcal{X} = \mathcal{X} \setminus \mathcal{S} \]

the associated split is $S \subseteq S$. In other words, for every cluster $C$ we define

in the complement of a cluster in which the associated splits directly

and \( q \in \mathcal{C} \) on $\mathcal{X} = \mathcal{X} \setminus \mathcal{C}$. We could simplify define the associated

by the clusters $\{ q \}$ on $\mathcal{X}$. For a given cluster $C$, we could simply define the associated

\[ \llbracket \frac{q}{q} \rrbracket = 0 \]

Now let us look at the opposite problem of defining a set of splits $S$ for a given

namely $\hat{a}$ and $\hat{b}$.

and clusters. On the other hand, if the root is chosen so as to subdivide some edge

in the rooted tree, then there is a unique one-to-one correspondence between the splits

of all clusters represented by the rooted version of $\mathcal{T}$. If we choose a node of $\mathcal{T}$

Let $\mathcal{S}$ be a set of splits on $\mathcal{X}$. If $\mathcal{X}$ is compatible, then there exists an unrooted

Spills and unrooted phylogenetic networks

92
are given the same color. Such a mapping is called a coloring. In this case, we say that the colors red and green are adjacent in the graph.

A graph is a mapping \( f: V \rightarrow \mathcal{C} \) where \( V \) is the vertex set of the graph, and \( \mathcal{C} \) is a set of colors. For a given graph, we say that two vertices are adjacent if their colors are distinct. A graph is said to be properly colored if no two adjacent vertices have the same color.

To develop the necessary formal concepts, let \( G = (V, E) \) be a connected graph. Below, we define a proper graph coloring with an arbitrary number of colors.

In general, there are \( k \) colors, where \( k \) is the number of vertices in the graph. For each color \( i \), we define a proper graph coloring by assigning a color \( c_i \) to each vertex in the graph.

When we obtain a graph by selecting a subgraph of the vertices by \( E \) and \( V \), and then define a graph coloring of the graph, we say that the resulting graph is properly colored.

We define the concept of a proper graph coloring with an arbitrary number of colors. For example, see Figure 4.4.

As we have seen, any set of compatible splits (containing all leaf splits) corresponds to a connected graph.

**Figure 4.4:** A graph network on \( X = \{a, \ldots, p\} \). Every leaf edge and, in this example, also the edge separating the leaves \( a \) and \( c \) from the rest of the tree, each represent a split in the same network.
\( d(\varepsilon') \leq d(\varepsilon) \) where \( \varepsilon' \) and \( \varepsilon \) are the new and old colors of vertices, respectively.

We consider two cases:

1. If \( d(\varepsilon') = d(\varepsilon) \), then we assign the same color to both vertices.
2. If \( d(\varepsilon') < d(\varepsilon) \), then we assign a different color to the new vertex.

This ensures that the coloring of the graph is proper, meaning no two adjacent vertices have the same color.
Consider an edge $e$ that has color $c$ and assume that $d = \delta$.

Proof: We show that $G$ has at most two components.

The proof statement implies that any two nodes $a$ and $w$ in the same

as claimed.

(5.12)

Putting these two observations together, we see that

$$
\{ (a, w) \} \cap (m, a) \subseteq \{ (a, w) \} \cap (m, a) = (m, a)
$$

Moreover, because $\text{len}(a, w) \geq 1$, we also have

$$
\{ (a, w) \} \cap (m, a) = (m, a)
$$

as claimed.

(5.5)

$$
\{ (a, w) \} \subseteq (m, a)^+ = (m, a) \subseteq (m, a)
$$

Now, because $\text{len}(a, w) \geq 1$, we can apply the induction

(5.8)

$$
\{ (m, a) \} \subseteq (m, a)
$$

path from $a$ to $m$, implying

In the first case every shortest path from $a$ to $m$ can be extended to a shortest

(5.2)

$$
1 + (m, a) = (m, a) \quad \text{or} \quad 1 - (m, a) = (m, a)
$$

must either have

\[ (m, a) = (m, a) \]

Because $(m, a) < 1$ for some $(m, a) = (m, a) < (d, a)$.

Now we can extend the induction assumption $(\text{len}(a, w)) = (d, a)$.

In this case is a shortest

By induction on $(\text{len}(a, w)) = (m, a)

First we show that any path $G$ from $a$ to $m$ uses all colors from $a$ to $b$. By
Define \( \sigma \rightarrow \rho \rightarrow \tau \leftarrow \zeta \rightarrow \xi \leftarrow \eta \rightarrow \mu \leftarrow \nu \rightarrow \omega \rightarrow \alpha \leftarrow \beta \rightarrow \gamma \rightarrow \delta \rightarrow \epsilon \rightarrow \zeta \rightarrow \xi \rightarrow \eta \rightarrow \mu \rightarrow \nu \rightarrow \omega \rightarrow \alpha \) is obvious that all edges of the graph of \( \Gamma \) and the other containing all nodes labeled with elements \( d \) of \( D \) are drawn parallel with the same length.

Note that any unrooted phylogenetic tree \( T \) can be regarded as a split network.

Figure 5.6: A split network \( \mathcal{N} \) representing all edges of color 1 produce a graph consisting of precisely two connected components, one containing all nodes labeled with elements of \( \mathcal{D} \). This implies that \( \Gamma \) has at most two components.

In every node is connected either to 0 or to \( m \), but not to both of these holds:

\[ (m, a) \in \mathcal{E} \setminus \mathcal{E} \setminus \mathcal{E} = \emptyset \]

Then, the implication as above that \( \mathcal{D} \) is an edge is drawn parallel with the same length. In this case, all edges represented are drawn parallel with the same length. However, the labeling of edges by splits is usually omitted, as shown in (a). Edges are labeled in (b) all edges represented are drawn parallel with the same length.

Figure 5.6: A split network \( \mathcal{N} \) representing all edges of color 1 produce a graph consisting of precisely two connected components, one containing all nodes labeled with elements of \( \mathcal{D} \). This implies that \( \Gamma \) has at most two components.

In every node is connected either to 0 or to \( m \), but not to both of these holds:

\[ (m, a) \in \mathcal{E} \setminus \mathcal{E} \setminus \mathcal{E} = \emptyset \]

Then, the implication as above that \( \mathcal{D} \) is an edge is drawn parallel with the same length. In this case, all edges represented are drawn parallel with the same length. However, the labeling of edges by splits is usually omitted, as shown in (a). Edges are labeled in (b) all edges represented are drawn parallel with the same length.
6.5 The canonical split network

Given by

\[ \{c, e, f, \alpha, \beta, \gamma, \delta, \epsilon, \zeta, \xi, \omega \} = \mathcal{X} \]

Exercise 5.5.5 (Draw a split network)

Consider the set of splits $\mathcal{S}$ on $\mathcal{X}$.

Let $\mathcal{S}$ be a set of splits on $\mathcal{X}$. The definition of a split network (Definition 5.3.3) gives

\[ \{c, e, f, \alpha, \beta, \gamma, \delta, \epsilon, \zeta, \xi, \omega \} \]

Figure 5.7 demonstrates that the split network associated with a set of splits is not unique.

Lemma 5.5.4 (Split networks and compatibility)

A set of splits $\mathcal{S}$ on $\mathcal{X}$ is compatible if and only if there exists a split network $N$ representing $\mathcal{S}$.

Definition of a compatible split network, split network, and P-adic number.

Theorem: A split network is a tree. The split network $N$ also represents $\mathcal{S}$, but it is not a split network for $\mathcal{X}$. Because $\mathcal{S}$ is compatible, there exists a corresponding split of the two non-trivial splits $\mathcal{S}$, together with all splits of the two non-trivial splits $\mathcal{S}$, shown in (a) and (b) (both represent the same set of splits $\mathcal{S}$).
As an example, consider the set of splits $\{p, q\}$, $\{r, s\}$, $\{t\}$, $\{u\}$, $\{v\}$, $\{w\}$. $\{p, q\}$ is an example of a set of splits that is not contained in any other set of splits.

For each axiom $x$ in $\mathcal{A}$, we set $\mathcal{Y}(x)$, where $\mathcal{Y}(x)$ denotes the unique projection which

exists.

If $p$ and $q$ are connected by an edge of color $\alpha \in \mathcal{E}$, any two nodes $p'$ and $p''$ are connected by an edge of color $\alpha \in \mathcal{E}$.

The network $\mathcal{G}$ is given by the set of all projections $\mathcal{P}(S)$ of $S$.

The network is defined as follows:

$\mathcal{G} = \{\mathcal{G}, \mathcal{E}, \mathcal{P}\}$

We can now define the canonical split network for a given set of splits $S$.

We have the following easy result:

Every axiom $x$ in $\mathcal{A}$ is contained in the core of exactly one projection of $S$.

Definition 7.6.9 (Canonical Split Network) Let $S$ be a set of splits on $X$. The canonical split network $\mathcal{G}$ is defined as follows:

- $\mathcal{G}$ is the set of all projections $\mathcal{P}(S)$ of $S$.
- $\mathcal{E}$ is the set of all edges $\{x, y\}$ such that $x$ and $y$ are connected in exactly one projection of $D$.

The core of a projection $D$ is defined as the set $\mathcal{Y}(x)$ of all $x$ for which $x \in \mathcal{Y}(D)$.

For all $x, y \in \mathcal{G}$, if $x \neq y$, then $\mathcal{Y}(x) \neq \mathcal{Y}(y)$.

The canonical split network $\mathcal{G}$ consists of the two split parts of the $i$-th split.

Properties:

- A vector $D$ of length $|D|$ is contained in at most $|D|$ projections.
- Let $S$ be a set of splits on $X$. We call the following

Definition 7.6.1 (Projection) Let $D$ be a set of splits on $X$. We call $D$
exist two projections and that differ only on their 1-connected graph. Let \( p \) be

\( \begin{align*}
&= 1, & L \in & \mathcal{X}, \text{ where } L \text{ contains an edge that color 1 is there, and}
&\text{where we first show that the edge coloring of } N \text{ is not unique. Consider any number}

\text{Buneman Graphs.}

\text{Let } S = \{ s_1, s_2, \ldots, s_n \} \text{ be a set of slips on } \mathcal{X} \text{ and let } N \text{ be the corresponding}

\text{Definition 5.2.1.}\text{ Then the graph } G_n \text{ is the map of Figure 5.2.3.} \text{ To prove this, we must show that the Buneman graph is a split graph, as defined in}

\text{The Buneman Graph for } S \text{ is a split network for } S.

\text{Lemma 5.6.5. Buneman graph is a split network.}\text{ Let } S = \{ s_1, s_2, \ldots, s_n \} \text{ be a set of slips on } \mathcal{X}.\text{ The}

\text{Exercise 5.6.4. Buneman graph and compatibility}\text{ Show that the Buneman graph}

\text{shown in Figure 5.6.8.}\text{ The resulting split network is}

\text{Thus the mapping of the network is}

\text{The consistent set of slips is an unrooted phylogenetic tree.}

\text{Example 5.6.5.}\text{ The Buneman graph associated with}

\text{The consistent set of projections are:}

\begin{align*}
&= \{ \{ p, q \}, \{ p, q, v \}, \{ p, q, v, w \} \} \text{ and } \{ \{ p, q \}, \{ p, q, v \}, \{ p, q, v, w \} \} \text{ and } \{ \{ p \}, \{ q \} \} \text{ and } \{ \{ \} \} \text{ and } \{ \{ \} \} \text{ and } \{ \{ \} \}

\end{align*}
exists a path from 0 to \( n \) in \( G \) with the desired properties.

This implies that \( D' \) is a valid projection and hence a node of the projection graph. Let \( G' \) be an induced subgraph of \( G \) with the same operation as \( G \). Then, the operation on \( G' \) is the same as the operation on \( G \). Hence, the operation on \( G' \) is also a valid projection.

Therefore, the operation on \( G' \) is a valid projection and hence a node of the projection graph. Let \( G'' \) be an induced subgraph of \( G' \) with the same operation as \( G' \). Then, the operation on \( G'' \) is the same as the operation on \( G' \). Hence, the operation on \( G'' \) is also a valid projection.

Therefore, the operation on \( G'' \) is a valid projection and hence a node of the projection graph. Let \( G''' \) be an induced subgraph of \( G'' \) with the same operation as \( G'' \). Then, the operation on \( G''' \) is the same as the operation on \( G'' \). Hence, the operation on \( G''' \) is also a valid projection.

Therefore, the operation on \( G''' \) is a valid projection and hence a node of the projection graph. Let \( G'''' \) be an induced subgraph of \( G''' \) with the same operation as \( G''' \). Then, the operation on \( G'''' \) is the same as the operation on \( G''' \). Hence, the operation on \( G'''' \) is also a valid projection.

Therefore, the operation on \( G'''' \) is a valid projection and hence a node of the projection graph. Let \( G''''' \) be an induced subgraph of \( G'''' \) with the same operation as \( G'''' \). Then, the operation on \( G''''' \) is the same as the operation on \( G'''' \). Hence, the operation on \( G''''' \) is also a valid projection.
5.6 The canonical split network

It remains to be shown that all shortest paths from \( v \) to \( w \) have the desired properties. To this end, consider an arbitrary path \( P \) of length \( d \) from \( p \) to \( q \) and let \( p' \) be the second node in the path. By assumption, \( p' \) is connected to \( q \) by a path containing \( d - 1 \) edges, each colored by a different index and none colored by the index of the component on which \( p \) and \( p' \) differ. Hence, \( P \) has \( d \) different colors.

Let us demonstrate now that \( N \) is bipartite. Chose a fixed taxon \( x \) in \( \mathcal{X} \). The parity of a projection \( p \) is set to 0, if the number of components of \( p \) that contain the taxon \( x \) is even, and is set to 1, otherwise. Using this measure, we can partition the set of nodes \( V \) of the Buneman graph in two disjoint subsets \( V_1 \) and \( V_2 \) such that for every edge \( e \in E \) one of the endpoints lies in \( V_1 \) and the other endpoint lies in \( V_2 \), in the following way: Let \( V_1 \) and \( V_2 \) be the set of nodes with odd and even parity, respectively. Then, for each pair of nodes \( u, v \) in \( V_1 \) (or in \( V_2 \)), there does not exist an edge connecting them since they differ on more than one component. This completes the proof that the Buneman graph is a split graph as defined in Definition 5.5.1.

To prove that the Buneman graph is a split network, we need to show for the node labeling \( \lambda : \mathcal{X} \rightarrow V \) and for every split \( S = A \cup B \) in \( \mathcal{S} \) that the deletion of all edges of color \( S \) produces a graph consisting of precisely two connected components, one containing all nodes labeled with elements of \( A \) and the other containing all nodes labeled with elements of \( B \) (see Definition 5.5.1). Theorem 5.5.2 ensures that, for each split \( S \) of a split set \( \mathcal{S} \) represented by \( N \), by deleting all edges of color \( S \) we obtain a graph \( N_S \) consisting of precisely two connected components \( N^0_S \) and \( N^1_S \). We need to prove that all nodes labeled with elements of \( A \) (or of \( B \)) are contained in \( N^0_S \) (or \( N^1_S \), respectively).

In the proof of Theorem 5.5.2, we showed that any two nodes \( p \) and \( q \) are contained in the same connected component of \( N_S \) if and only if \( S \notin \sigma(p, q) \) holds. For every pair of nodes \( p \) and \( q \) such that \( \lambda^{-1}(p) = a \) and \( \lambda^{-1}(q) = b \), with \( a \) in \( A \) and \( b \) in \( B \), it holds that \( p \) and \( q \) differ on the split \( S \). As demonstrated above, this implies that \( S \in \sigma(p, q) \), so \( p \) and \( q \) are not in the same connected component of \( N^0_S \) and \( N^1_S \). Since this holds for each pair of nodes \( p \) and \( q \) in \( V \) and for every split, this concludes the proof.

Let \( \mathcal{S} \) be a set of splits on \( \mathcal{X} \). How many nodes and edges might the canonical split network contain? If the incompatibility graph \( IG(S) \) contains a clique of size \( k \), then any one of the \( 2^k \) possible choices of split parts gives rise to a node in the network and thus the number of nodes and edges of the network is exponential in the number of splits in the worst case.

In Chapter 7 we discuss how to compute the Buneman graph using the convex hull algorithm.
Definition 5.7.1 (Circular spils) A set of spils $S$ on $X$ is called circular, if

exits a linear ordering $(x_1, \ldots, x_n)$ of the elements of $X$ such that each

$circle$ is convex. More formally, a set of spils $S$ is a non-planar network if

and the corresponding spils network is shown in Figure 5.7. An example of a

and the other containing all edges in $S$ (see Figure 5.9). For example, the

the edges that separate the two half-planar one containing all edges in $S$ can be realized by a line through

around a circle in such a way that each spil $s \in S$ can be realized by a line through

Importantly, a set of spils $S$ on $X$ is called circular if the edges in $S$ can be drawn

which we introduce in the next section.

Circular spils, which are the focus of this section, and weakly connected

circular spils, which are the focus of this section, and weakly connected

in an attempt to avoid overly complicated networks. The two most important

way. Hence, a number of restricted classes of sets of spils have been introduced

networks can be very complicated and thus difficult to visualize in a comprehensible

One practical problem that arises when working with spil networks is that the

5.7 Circular spils and planar spil networks

Figure 5.10 (a) A set of four non-circular spils $S$ on $X$.

(a) Non-planar network

(b) Circular ordering

(c) An outer-hinged planar spil network

that separates the two half-planar one containing all edges in $S$ can be realized by a straight line through

d the other containing all edges in $S$ can be realized by a straight line through

d around a circle in such a way that each spil $s \in S$ can be realized by a line through

dImportantly, a set of spils $S$ on $X$ is called circular if the edges in $S$ can be drawn

which we introduce in the next section.

Circular spils, which are the focus of this section, and weakly connected

circular spils, which are the focus of this section, and weakly connected

in an attempt to avoid overly complicated networks. The two most important

way. Hence, a number of restricted classes of sets of spils have been introduced

networks can be very complicated and thus difficult to visualize in a comprehensible

One practical problem that arises when working with spil networks is that the

Spils and unrooted phylogenetic networks