Advanced Algorithms

Rearrangement Distance of Phylogenetic Trees

kernelization, fpt, approximation algorithm

Johannes Zink · WS22
Phylogenetic Trees

... represent the evolutionary history of a set of taxa.

Kingfishers (German: *Eisvögel*)
by McCullough et al. (2016)
Phylogenetic Trees

... represent the evolutionary history of a set of taxa.

Tree of Life
www.evogeneao.com
(2017)
Phylogenetic Trees

... represent the evolutionary history of a set of taxa.

Tree of Life
www.evogeneao.com
(2017)

Phylogenetic tree of the Indo-European languages
by Chang & Chundra
(2015)
Phylogenetic Trees

... represent the evolutionary history of a set of taxa.

Properties (in the biological sense):
Phylogenetic Trees

... represent the evolutionary history of a set of taxa.

Properties (in the biological sense):

- Leaves are labelled with taxa.
- Each taxon represents a species, population, individual organism, gene, chromosome, ...
Phylogenetic Trees

... represent the evolutionary history of a set of taxa.

Properties (in the biological sense):

- Leaves are labelled with taxa.
- Each taxon represents a species, population, individual organism, gene, chromosome, ...
- Edge length represents an amount of time passed or a genetic distance.
Phylogenetic Trees

... represent the evolutionary history of a set of taxa.

**Properties** (in the biological sense):

- Leaves are labelled with taxa.
- Each taxon represents a species, population, individual organism, gene, chromosome, ...
- Edge length represents an amount of time passed or a genetic distance.
- Inference methods compute a phylogenetic tree based on some model and data.

Kingfishers (German: *Eisvögel*) by McCullough et al. (2016)
Phylogenetic Trees

Let $X = \{1, 2, 3, \ldots, n\}$.

A (rooted, binary) phylogenetic tree $T$ is a rooted tree with the following properties:

- The unique root is labeled $\rho$ and has outdegree 1.
- The leaves are bijectively labeled by $X$.
- All other vertices have indegree 1 and outdegree 2 (i.e., it is a binary tree).
Phylogenetic Trees

Let $X = \{1, 2, 3, \ldots, n\}$. A (rooted, binary) phylogenetic tree $T$ is a rooted tree with the following properties:

- The unique root is labeled $\rho$ and has outdegree 1.
- The leaves are bijectively labeled by $X$.
- All other vertices have indegree 1 and outdegree 2 (i.e., it is a binary tree).
Phylogenetic Trees

Let $X = \{1, 2, 3, \ldots, n\}$.

A (rooted, binary) phylogenetic tree $T$ is a rooted tree with the following properties:

- The unique root is labeled $\rho$ and has outdegree 1.
- The leaves are bijectively labeled by $X$.
Phylogenetic Trees

Let $X = \{1, 2, 3, \ldots, n\}$.

A (rooted, binary) phylogenetic tree $T$ is a rooted tree with the following properties:

- The unique root is labeled $\rho$ and has outdegree 1.
- The leaves are bijectively labeled by $X$.
- All other vertices have indegree 1 and outdegree 2 (i.e., it is a binary tree).
Phylogenetic Trees

Let $X = \{1, 2, 3, \ldots, n\}$.

A (rooted, binary) phylogenetic tree $T$ is a rooted tree with the following properties:

- The unique root is labeled $\rho$ and has outdegree 1.
- The leaves are bijectively labeled by $X$.
- All other vertices have indegree 1 and outdegree 2 (i.e., it is a binary tree).

Remarks. Here, in our definition

- vertices have no heights and
- the order of the children of a vertex does not matter.
Problem

For the same taxa, we may infer different phylogenetic trees because of the use of
- different inference methods,
- different models, or
- different data.
Problem

For the same taxa, we may infer different phylogenetic trees because of the use of

- different inference methods,
- different models, or
- different data.

We want to be able to compare different phylogenetic trees. How?
Problem

For the same taxa, we may infer **different** phylogenetic trees because of the use of
- different inference methods,
- different models, or
- different data.

We want to be able to **compare** different phylogenetic trees. How?

**Goal.**

Define a **metric** that specifies how similar two phylogenetic trees on the same set $X$ are and devise algorithms to compute it.
Problem

For the same taxa, we may infer different phylogenetic trees because of the use of:
- different inference methods,
- different models, or
- different data.

We want to be able to compare different phylogenetic trees. How?

Goal.
Define a metric that specifies how similar two phylogenetic trees on the same set $X$ are and devise algorithms to compute it.

**Definition:**
A metric $d$ is a function of two parameters such that:
- $d(x, x) = 0$ (no distance to itself)
- $d(x, y) > 0$ for $x \neq y$ (positive)
- $d(x, y) = d(y, x)$ (symmetric)
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds)
Problem

For the same taxa, we may infer different phylogenetic trees because of the use of
■ different inference methods,
■ different models, or
■ different data.

We want to be able to compare different phylogenetic trees. How?

Goal.
Define a metric that specifies how similar two phylogenetic trees on the same set \( X \) are and devise algorithms to compute it.

Idea.
Count the number of rearrangement operations that are necessary to transform \( T \) into \( T' \).
**Subtree Prune & Regraft (SPR)**

An **SPR** operation transforms one phylogenetic tree into another one.

\[ T \]

\[ \rho \]

1 2 3 4 5
Subtree Prune & Regraft (SPR)

An **SPR** operation transforms one phylogenetic tree into another one.
Subtree Prune & Regraft (SPR)

An SPR operation transforms one phylogenetic tree into another one.
**Subtree Prune & Regraft (SPR)**

An **SPR** operation transforms one phylogenetic tree into another one.

![Diagram of Subtree Prune & Regraft (SPR)]
**Subtree Prune & Regraft (SPR)**

An **SPR** operation transforms one phylogenetic tree into another one.

![Diagram](attachment:image.png)
Subtree Prune & Regraft (SPR)

An **SPR** operation transforms one phylogenetic tree into another one.

![Diagram demonstrating the SPR operation](image)

Subtree pruning and regrafting.
**Subtree Prune & Regraft (SPR)**

An **SPR** operation transforms one phylogenetic tree into another one.

---

**Diagram:**

- **Original Tree ($T$):**
  - Node $\rho$,
  - Subtree 1, 2, 3,
  - Nodes 4, 5.

- **SPR Operation:**
  - Prune subtree 1, 2, 3.
  - Regraft 4, 5.

- **Transformed Tree ($T'$):**
  - Node $\rho$,
  - Nodes 1, 4, 5,
Subtree Prune & Regraft (SPR)

An SPR operation transforms one phylogenetic tree into another one.

Note that an SPR operation is reversible.
SPR-Graph

The SPR operations induce the **SPR-graph** \( G = (V, E) \) for a set \( X \):
The SPR operations induce the **SPR-graph** $G = (V, E)$ for a set $X$:

- $V = \{ T \mid T \text{ is a phylogenetic tree on } X \}$
The SPR operations induce the **SPR-graph** $G = (V, E)$ for a set $X$:

- $V = \{T \mid T$ is a phylogenetic tree on $X\}$
- $E = \{\{T, T'\} \mid T$ can be transformed into $T'$ with a single SPR operation\}$
The **SPR-distance** $d_{\text{SPR}}(T, T')$ of $T$ and $T'$ is defined as the distance of $T$ and $T'$ in the SPR-graph $G$. 

---
The **SPR-distance** $d_{\text{SPR}}(T, T')$ of $T$ and $T'$ is defined as the distance of $T$ and $T'$ in the SPR-graph $G$.

**Lemma 1.**
The SPR-graph $G$ is connected.
The **SPR-distance** $d_{\text{SPR}}(T, T')$ of $T$ and $T'$ is defined as the distance of $T$ and $T'$ in the SPR-graph $G$.

**Lemma 1.**
The SPR-graph $G$ is connected.

**Proof** exercise
SPR-Distance

The **SPR-distance** $d_{SPR}(T, T')$ of $T$ and $T'$ is defined as the distance of $T$ and $T'$ in the SPR-graph $G$.

**Lemma 1.**
The SPR-graph $G$ is connected.

**Proof** exercise

**Lemma 2.**
The SPR-distance is a metric.
The **SPR-distance** $d_{\text{SPR}}(T, T')$ of $T$ and $T'$ is defined as the distance of $T$ and $T'$ in the SPR-graph $G$.

**Lemma 1.**
The SPR-graph $G$ is connected.

**Lemma 2.**
The SPR-distance is a metric.

**Definition:**
A metric $d$ is a function of two parameters such that:
- $d(x, x) = 0$ (no distance to itself)
- $d(x, y) > 0$ for $x \neq y$ (positive)
- $d(x, y) = d(y, x)$ (symmetric)
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds)
The **SPR-distance** $d_{SPR}(T, T')$ of $T$ and $T'$ is defined as the distance of $T$ and $T'$ in the SPR-graph $G$.

**Lemma 1.**
The SPR-graph $G$ is connected.

**Lemma 2.**
The SPR-distance is a metric.

**Proof.** $G$ is connected and undirected.

**Definition:**
A *metric* $d$ is a function of two parameters such that:
- $d(x, x) = 0$ (no distance to itself)
- $d(x, y) > 0$ for $x \neq y$ (positive)
- $d(x, y) = d(y, x)$ (symmetric)
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds)
SPR-Distance

The **SPR-distance** $d_{\text{SPR}}(T, T')$ of $T$ and $T'$ is defined as the distance of $T$ and $T'$ in the SPR-graph $G$.

**Lemma 1.**
The SP2R-graph $G$ is connected.

**Lemma 2.**
The SPR-distance is a metric.

**Proof.** $G$ is connected and undirected.

**Definition:**
A *metric* $d$ is a function of two parameters such that:
- $d(x, x) = 0$ (no distance to itself)
- $d(x, y) > 0$ for $x \neq y$ (positive)
- $d(x, y) = d(y, x)$ (symmetric)
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds)
SPR-Distance

The **SPR-distance** $d_{SPR}(T, T')$ of $T$ and $T'$ is defined as the distance of $T$ and $T'$ in the SPR-graph $G$.

**Lemma 1.**
The SPR-graph $G$ is connected.

**Lemma 2.**
The SPR-distance is a metric.

**Proof.** $G$ is connected and undirected.

**Definition:**
A *metric* $d$ is a function of two parameters such that:

- $d(x, x) = 0$ (no distance to itself)
- $d(x, y) > 0$ for $x \neq y$ (positive)
- $d(x, y) = d(y, x)$ (symmetric)
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds)

*✓* trivial

*✓* shortest path exists because $G$ is connected
SPR-Distance

The **SPR-distance** $d_{SPR}(T, T')$ of $T$ and $T'$ is defined as the distance of $T$ and $T'$ in the SPR-graph $G$.

**Lemma 1.**
The SPR-graph $G$ is connected.

**Lemma 2.**
The SPR-distance is a metric.

**Proof.** $G$ is connected and undirected.

**Definition:**
A **metric** $d$ is a function of two parameters such that:
- $d(x, x) = 0$ (no distance to itself)
- $d(x, y) > 0$ for $x \neq y$ (positive)
- $d(x, y) = d(y, x)$ (symmetric)
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds)

- trivial
- shortest path exists because $G$ is connected
- all paths can be reversed bc. $G$ is undirected
The **SPR-distance** $d_{\text{SPR}}(T, T')$ of $T$ and $T'$ is defined as the distance of $T$ and $T'$ in the SPR-graph $G$.

**Lemma 1.**
The SPR-graph $G$ is connected.

**Lemma 2.**
The SPR-distance is a metric.

**Proof.** $G$ is connected and undirected.

**Definition:**
A **metric** $d$ is a function of two parameters such that:

- $d(x, x) = 0$ (no distance to itself)  
- $d(x, y) > 0$ for $x \neq y$ (positive)  
- $d(x, y) = d(y, x)$ (symmetric)  
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds)

- trivial
- shortest path exists because $G$ is connected
- all paths can be reversed bc. $G$ is undirected
- the triangle inequality holds because we can compose the path $x \rightsquigarrow z$ by $x \rightsquigarrow y \rightsquigarrow z$
SPR-Distance

The **SPR-distance** $d_{\text{SPR}}(T, T')$ of $T$ and $T'$ is defined as the distance of $T$ and $T'$ in the SPR-graph $G$.

**Lemma 1.**
The SPR-graph $G$ is connected.

**Defintion:**
A *metric* $d$ is a function of two parameters such that:
- $d(x, x) = 0$ (no distance to itself)
- $d(x, y) > 0$ for $x \neq y$ (positive)
- $d(x, y) = d(y, x)$ (symmetric)
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality holds)

**Lemma 2.**
The SPR-distance is a metric.

**Proof.** $G$ is connected and undirected.
All properties of a metric follow.

- trivial
- shortest path exists because $G$ is connected
- all paths can be reversed bc. $G$ is undirected
- the triangle inequality holds because we can compose the path $x \leftrightarrow z$ by $x \leftrightarrow y \leftrightarrow z$
**SPR-Distance**

The **SPR-distance** $d_{SPR}(T, T')$ of $T$ and $T'$ is defined as the distance of $T$ and $T'$ in the SPR-graph $G$.

**Lemma 1.**
The SPR-graph $G$ is connected.

**Proof exercise**

**Goal.**
Compute the SPR-distance $d_{SPR}(T, T')$.

**Lemma 2.**
The SPR-distance is a metric.

**Proof.** $G$ is connected and undirected. All properties of a metric follow.
The **SPR-distance** $d_{\text{SPR}}(T, T')$ of $T$ and $T'$ is defined as the distance of $T$ and $T'$ in the SPR-graph $G$.

**Lemma 1.**
The SPR-graph $G$ is connected.

**Proof** exercise

**Goal.**
Compute the SPR-distance $d_{\text{SPR}}(T, T')$.

...but $G$ is huge!

$$|V(G)| = (2n - 3)!! = (2n - 3) \cdot (2n - 5) \cdot \ldots \cdot 5 \cdot 3$$

**Lemma 2.**
The SPR-distance is a metric.

**Proof.** $G$ is connected and undirected. All properties of a metric follow.

$\square$
SPR-Distance

The SPR-distance \( d_{\text{SPR}}(T, T') \) of \( T \) and \( T' \) is defined as the distance of \( T \) and \( T' \) in the SPR-graph \( G \).

**Lemma 1.** The SPR-graph \( G \) is connected.

**Proof** exercise

**Lemma 2.** The SPR-distance is a metric.

**Proof.** \( G \) is connected and undirected. All properties of a metric follow.

...but \( G \) is huge!

\[
|V(G)| = (2n - 3)!! = (2n - 3) \cdot (2n - 5) \cdot \ldots \cdot 5 \cdot 3
\]

Can we rephrase the problem?
Maximum Agreement Forests

\[
\begin{array}{ccc}
T & \xrightarrow{\text{SPR}} & \hat{T} \\
1 & 2 & 3 & 4 & 5 \\
\end{array}
\]

\[
\begin{array}{ccc}
\hat{T} & \xrightarrow{\text{SPR}} & T' \\
1 & 2 & 3 & 4 & 5 \\
\end{array}
\]
Maximum Agreement Forests

\[ T \xrightarrow{\text{SPR}} \hat{T} \xrightarrow{\text{SPR}} T' \]

\[ \rho \]

\[ 1 \ 2 \ 3 \ 4 \ 5 \]

\[ \hat{T} \]

\[ \rho \]

\[ 1 \ 4 \ 2 \ 3 \ 5 \]

\[ T' \]

\[ \rho \]

\[ 4 \ 1 \ 2 \ 3 \ 5 \]

\[ \hat{T} \]

\[ \rho \]

\[ 1 \ 2 \ 3 \ 4 \ 5 \]

\[ \rho \]
Maximum Agreement Forests

\[ T \xrightarrow{\text{SPR}} \hat{T} \xrightarrow{\text{SPR}} T' \]
Maximum Agreement Forests

\[ T \xrightarrow{\text{SPR}} \hat{T} \xrightarrow{\text{SPR}} T' \]

\[ F \]

\[ 1 \ 2 \ 3 \ 4 \ 5 \]

\[ 1 \ 4 \ 2 \ 3 \ 5 \]

\[ 4 \ 1 \ 2 \ 3 \ 5 \]

\[ 1 \ 5 \]

\[ 2 \ 3 \]

\[ 4 \]
Maximum Agreement Forests

\[ T \xrightarrow{\text{SPR}} \hat{T} \xrightarrow{\text{SPR}} T' \]

\[ F \text{ into } T \]

\[ F \]

\[ \rho \]

\[ 1 \quad 2 \quad 3 \quad 4 \quad 5 \]

\[ \rho \]

\[ 1 \quad 5 \]

\[ 2 \quad 3 \]

\[ 4 \]
Maximum Agreement Forests

\[ T \xrightarrow{\text{SPR}} \hat{T} \xrightarrow{\text{SPR}} T' \]

\[ F \text{ into } T \]

\[ F \]

\[ F \text{ into } T' \]
An agreement forest (AF) $F$ of $T$ and $T'$ is a forest $\{T_{\rho}, T_1, T_2, \ldots, T_k\}$ such that

- the label sets of the $T_i$ partition $X \cup \{\rho\}$,
Maximum Agreement Forests

An agreement forest (AF) $F$ of $T$ and $T'$ is a forest $\{T_\rho, T_1, T_2, \ldots, T_k\}$ such that

- the label sets of the $T_i$ partition $X \cup \{\rho\}$,
- $\rho$ is in the label set of $T_\rho$, and
An agreement forest (AF) $F$ of $T$ and $T'$ is a forest \{ $T_\rho$, $T_1$, $T_2$, \ldots, $T_k$ \} such that

- the label sets of the $T_i$ partition $X \cup \{ \rho \}$,
- $\rho$ is in the label set of $T_\rho$, and
- there is an edge-disjoint embedding of the $T_i$s into $T$ and $T'$ where all edges of $T$ and $T'$ are covered. In other words, we can place all $T_i$s onto $T$ and $T'$ such that the $T_i$s do not overlap and every edge of $T$ and $T'$ lies under some $T_i$. 

---

Maximum Agreement Forests
Maximum Agreement Forests

An agreement forest (AF) $F$ of $T$ and $T'$ is a forest $\{T_\rho, T_1, T_2, \ldots, T_k\}$ such that

- the label sets of the $T_i$ partition $X \cup \{\rho\}$,
- $\rho$ is in the label set of $T_\rho$, and
- there is an edge-disjoint embedding of the $T_i$s into $T$ and $T'$ where all edges of $T$ and $T'$ are covered. In other words, we can place all $T_i$s onto $T$ and $T'$ such that the $T_i$s do not overlap and every edge of $T$ and $T'$ lies under some $T_i$.

If $k$ is minimum, $F$ is a maximum agreement forest (MAF).
Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let
$F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$.
Define
$$m(T, T') = k = |F| - 1.$$
Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$. Define

$$m(T, T') = k = |F| - 1.$$  

**Theorem 3.** $m(T, T') = d_{\text{SPR}}(T, T')$
Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$. Define

$$m(T, T') = k = |F| - 1.$$ 

**Theorem 3.** $m(T, T') = d_{SPR}(T, T')$

**Proof** of “≤” by induction on $d = d_{SPR}(T, T')$. 

Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$.
Define
\[ m(T, T') = k = |F| - 1. \]

**Theorem 3.** $m(T, T') = d_{SPR}(T, T')$

**Proof** of “$\leq$” by induction on $d = d_{SPR}(T, T')$.
- Case $d = 0$ is trivial and Case $d = 1$ is easy. ✓
Characterization

Let \( T \) and \( T' \) be two phylogenetic trees on \( X \) and let \( F = \{T_\rho, T_1, T_2, \ldots, T_k\} \) be a MAF of \( T \) and \( T' \). Define

\[
m(T, T') = k = |F| - 1.
\]

**Theorem 3.** \( m(T, T') = d_{\text{SPR}}(T, T') \)

**Proof** of “\( \leq \)” by induction on \( d = d_{\text{SPR}}(T, T') \).
- Case \( d = 0 \) is trivial and Case \( d = 1 \) is easy. ✓
- Assume \( m(T, T') \leq d_{\text{SPR}}(T, T') \) holds for all \( d \leq \ell \).
Characterization

Let \( T \) and \( T' \) be two phylogenetic trees on \( X \) and let \( F = \{ T\rho, T_1, T_2, \ldots, T_k \} \) be a MAF of \( T \) and \( T' \). Define

\[
m(T, T') = k = |F| - 1.
\]

**Theorem 3.** \( m(T, T') = d_{\text{SPR}}(T, T') \)

**Proof** of “\( \leq \)” by induction on \( d = d_{\text{SPR}}(T, T') \).

- If \( d = \ell + 1 \), then there exists \( \hat{T} \) with \( d_{\text{SPR}}(T, \hat{T}) = \ell \) and \( d_{\text{SPR}}(\hat{T}, T') = 1 \).
Characterization

Let \( T \) and \( T' \) be two phylogenetic trees on \( X \) and let \( F = \{ T_\rho, T_1, T_2, \ldots, T_k \} \) be a MAF of \( T \) and \( T' \).

Define

\[
m(T, T') = k = |F| - 1.
\]

**Theorem 3.** \( m(T, T') = d_{\text{SPR}}(T, T') \)

**Proof** of “\( \leq \)” by induction on \( d = d_{\text{SPR}}(T, T') \).

- If \( d = \ell + 1 \), then there exists \( \hat{T} \) with \( d_{\text{SPR}}(T, \hat{T}) = \ell \) and \( d_{\text{SPR}}(\hat{T}, T') = 1 \).

\[
\begin{align*}
T \quad \text{\( \ell \) SPR} \quad \hat{T} \quad \hat{F} \quad \hat{T} \quad T' \quad F'
\end{align*}
\]
Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$. Define

$$m(T, T') = k = |F| - 1.$$ 

**Theorem 3.** $m(T, T') = d_{\text{SPR}}(T, T')$

**Proof** of “≤” by induction on $d = d_{\text{SPR}}(T, T')$. 

- If $d = \ell + 1$, then there exists $\hat{T}$ with $d_{\text{SPR}}(T, \hat{T}) = \ell$ and $d_{\text{SPR}}(\hat{T}, T') = 1$.

- $\exists$ MAF $\hat{F}$ for $T$ & $\hat{T}$ of size $\ell + 1$ and MAF $F'$ for $\hat{T}$ & $T'$ of size 2.

- Compose $\hat{T}$ by subtrees of $\hat{F}$. The subtree $T'_1$ of $F'$ is rooted at one edge of $\hat{T}$ within one subtree of $\hat{F}$.

\[ T \quad \hat{T} \quad \hat{F} \quad \hat{T} \quad T' \quad F' \]

\[ \rho \quad \rho \quad \rho \quad \rho \quad \rho \quad \rho \]
Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$. Define

$$m(T, T') = k = |F| - 1.$$ 

**Theorem 3.** $m(T, T') = d_{\text{SPR}}(T, T')$

**Proof** of “$\leq$” by induction on $d = d_{\text{SPR}}(T, T')$.

- If $d = \ell + 1$, then there exists $\hat{T}$ with $d_{\text{SPR}}(T, \hat{T}) = \ell$ and $d_{\text{SPR}}(\hat{T}, T') = 1$.

- $\exists$ MAF $\hat{F}$ for $T \& \hat{T}$ of size $\ell + 1$ and MAF $F'$ for $\hat{T} \& T'$ of size 2.

- Compose $\hat{T}$ by subtrees of $\hat{F}$. The subtree $T'_1$ of $F'$ is rooted at one edge of $\hat{T}$ within one subtree of $\hat{F}$. 

---

![Diagram](image.png)
Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$. Define

$$m(T, T') = k = |F| - 1.$$

**Theorem 3.** $m(T, T') = d_{\text{SPR}}(T, T')$

**Proof** of “$\leq$” by induction on $d = d_{\text{SPR}}(T, T')$.

- If $d = \ell + 1$, then there exists $\hat{T}$ with $d_{\text{SPR}}(T, \hat{T}) = \ell$ and $d_{\text{SPR}}(\hat{T}, T') = 1$.
- There exists MAF $\hat{F}$ for $T$ & $\hat{T}$ of size $\ell + 1$ and MAF $F'$ for $\hat{T}$ & $T'$ of size 2.
- Compose $\hat{T}$ by subtrees of $\hat{F}$. The subtree $T'_1$ of $F'$ is rooted at one edge of $\hat{T}$ within one subtree of $\hat{F}$.
- Subdivide the corresponding tree to obtain $F$ from $\hat{F}$, which is an AF for $T$ and $T'$. 
Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$.
Define
$$m(T, T') = k = |F| - 1.$$ 

**Theorem 3.** $m(T, T') = d_{SPR}(T, T')$

**Proof** of “$\geq$” by induction on $m = m(T, T')$. 


Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$. Define

$$m(T, T') = k = |F| - 1.$$ 

**Theorem 3.** $m(T, T') = d_{\text{SPR}}(T, T')$

**Proof** of “$\geq$” by induction on $m = m(T, T')$.

- Case $m = 0$ is trivial and Case $m = 1$ is easy. ✓
Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$. Define

$$m(T, T') = k = |F| - 1.$$

**Theorem 3.** $m(T, T') = d_{SPR}(T, T')$

**Proof** of “$\geq$” by induction on $m = m(T, T')$.
- Case $m = 0$ is trivial and Case $m = 1$ is easy.

![Diagram showing trees and MAF]
Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$. Define

$$m(T, T') = k = |F| - 1.$$

**Theorem 3.** \(m(T, T') = d_{\text{SPR}}(T, T')\)

**Proof** of “≥” by induction on \(m = m(T, T')\).

- Case \(m = 0\) is trivial and Case \(m = 1\) is easy.
- Assume \(m(T, T') \geq d_{\text{SPR}}(T, T')\) holds for all \(m \leq \ell\).
Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$. Define

$$m(T, T') = k = |F| - 1.$$ 

**Theorem 3.** $m(T, T') = d_{\text{SPR}}(T, T')$

**Proof** of “$\geq$” by induction on $m = m(T, T')$.

- Let $F$ be a MAF of $T$ and $T'$ of size $\ell + 2$. $\Rightarrow m = \ell + 1$
Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_{\rho}, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$. Define

$$m(T, T') = k = |F| - 1.$$ 

**Theorem 3.** $m(T, T') = d_{SPR}(T, T')$

**Proof** of “$\geq$” by induction on $m = m(T, T')$.

- Let $F$ be a MAF of $T$ and $T'$ of size $\ell + 2$. $\Rightarrow m = \ell + 1$
- There exists a $T_i$ that can be pruned in $T$ due to the nesting structure of subtrees.

\[ T \quad T' \quad F \]

\[
\begin{array}{c}
\rho \\
\end{array} \quad \begin{array}{c}
\rho \\
\end{array} \quad \begin{array}{c}
\rho \\
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\rho \\
\end{array} \\
\begin{array}{c}
\rho \\
\end{array} \\
\end{array} \quad \begin{array}{c}
\begin{array}{c}
\rho \\
\end{array} \\
\begin{array}{c}
\rho \\
\end{array} \\
\end{array} \quad \begin{array}{c}
\begin{array}{c}
\rho \\
\end{array} \\
\begin{array}{c}
\rho \\
\end{array} \\
\end{array}
\]


Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$.
Define

$$m(T, T') = k = |F| - 1.$$ 

**Theorem 3.** $m(T, T') = d_{SPR}(T, T')$

**Proof** of “$\geq$” by induction on $m = m(T, T')$.

- Let $F$ be a MAF of $T$ and $T'$ of size $\ell + 2$.
- There exists a $T_i$ that can be pruned in $T$ due to the nesting structure of subtrees.

![Diagram of trees and MAF](image)

- Regraft $T_i$ according to the embedding of $F$ into $T' \Rightarrow \hat{T} \& \hat{F}$
Characterization

Let \( T \) and \( T' \) be two phylogenetic trees on \( X \) and let \( F = \{ T_\rho, T_1, T_2, \ldots, T_k \} \) be a MAF of \( T \) and \( T' \). Define

\[
m(T, T') = k = |F| - 1.
\]

**Theorem 3.** \( m(T, T') = d_{\text{SPR}}(T, T') \)

**Proof** of “\( \geq \)” by induction on \( m = m(T, T') \).

- Let \( F \) be a MAF of \( T \) and \( T' \) of size \( \ell + 2 \).
- There exists a \( T_i \) that can be pruned in \( T \) due to the nesting structure of subtrees.

\[
\begin{array}{cccccc}
T & & T' & & F & & \hat{T} & & \hat{F} \\
\rho & & \rho & & \rho & & \rho & & \rho
\end{array}
\]

- Regraft \( T_i \) according to the embedding of \( F \) into \( T' \) \( \Rightarrow \) \( \hat{T} \& \hat{F} \)
- \( \hat{F} \) is AF for \( \hat{T} \& T' \) and \( |\hat{F}| = \ell + 1 \)
- \( \Rightarrow d_{\text{SPR}}(\hat{T}, T') \leq \ell \)
Characterization

Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$.
Define
\[
m(T, T') = k = |F| - 1.
\]

**Theorem 3.** $m(T, T') = d_{\text{SPR}}(T, T')$

**Proof** of “$\geq$” by induction on $m = m(T, T')$.

- Let $F$ be a MAF of $T$ and $T'$ of size $\ell + 2$.
- There exists a $T_i$ that can be pruned in $T$ due to the nesting structure of subtrees.

```
T      T'      F      \hat{T}    \hat{F}
\rho    \rho    \rho\triangle \rho\triangle
```

- Regraft $T_i$ according to the embedding of $F$ into $T'$ $\Rightarrow \hat{T}$ & $\hat{F}$
- $\hat{F}$ is AF for $\hat{T}$ & $T'$ and $|\hat{F}| = \ell + 1$
- $\Rightarrow d_{\text{SPR}}(\hat{T}, T') \leq \ell$
- $d_{\text{SPR}}(T, \hat{T}) = 1$
Let $T$ and $T'$ be two phylogenetic trees on $X$ and let $F = \{T_\rho, T_1, T_2, \ldots, T_k\}$ be a MAF of $T$ and $T'$.
Define

$$m(T, T') = k = |F| - 1.$$ 

**Theorem 3.** $m(T, T') = d_{\text{SPR}}(T, T')$

**Proof** of “$\geq$” by induction on $m = m(T, T')$.

- Let $F$ be a MAF of $T$ and $T'$ of size $\ell + 2$.
- There exists a $T_i$ that can be pruned in $T$ due to the nesting structure of subtrees.

![Diagram](image)

- Regraft $T_i$ according to the embedding of $F$ into $T' \Rightarrow \hat{T} & \hat{F}$
- $\hat{F}$ is AF for $\hat{T} & T'$ and $|\hat{F}| = \ell + 1$
- $\Rightarrow d_{\text{SPR}}(\hat{T}, T') \leq \ell$
- $d_{\text{SPR}}(T, \hat{T}) = 1$
- $d_{\text{SPR}}(T, T') \leq \ell + 1 = m(T, T')$
Theorem 4. [HJWZ ’96, BS ’05]
Computing $d_{\text{SPR}}(T, T')$ is NP-hard.

Proof by reduction from Exact Cover by 3-Sets.
Theorem 4. [HJWZ ’96, BS ’05]
Computing \( d_{\text{SPR}}(T, T') \) is NP-hard.

Proof by reduction from Exact Cover by 3-Sets.

Plan.
- Construct kernel of the problem.
- Replace \( T \) and \( T' \) with smaller \( S \) and \( S' \).
- Derive \( d_{\text{SPR}}(T, T') \) from \( d_{\text{SPR}}(S, S') \).
Theorem 4. [HJWZ ’96, BS ’05]
Computing $d_{SPR}(T, T')$ is NP-hard.

**Proof** by reduction from Exact Cover by 3-Sets.

**Plan.**
- Construct **kernel** of the problem.
  - Replace $T$ and $T'$ with smaller $S$ and $S'$.
  - Derive $d_{SPR}(T, T')$ from $d_{SPR}(S, S')$.
- Show that the size of the kernel depends on $d_{SPR}(T, T')$. 
Theorem 4. [HJWZ '96, BS '05] Computing $d_{\text{SPR}}(T, T')$ is NP-hard.

**Proof** by reduction from Exact Cover by 3-Sets.

**Plan.**

- Construct **kernel** of the problem.
  
  - Replace $T$ and $T'$ with smaller $S$ and $S'$.
  
  - Derive $d_{\text{SPR}}(T, T')$ from $d_{\text{SPR}}(S, S')$.

- Show that the size of the kernel depends on $d_{\text{SPR}}(T, T')$.

- Devise an FPT algorithm with respect to $d_{\text{SPR}}$. 
Theorem 4. [HJWZ ’96, BS ’05]
Computing $d_{\text{SPR}}(T, T')$ is NP-hard.

Proof by reduction from Exact Cover by 3-Sets.

Plan.

- Construct **kernel** of the problem.
  - Replace $T$ and $T'$ with smaller $S$ and $S'$.
  - Derive $d_{\text{SPR}}(T, T')$ from $d_{\text{SPR}}(S, S')$.
- Show that the size of the kernel depends on $d_{\text{SPR}}(T, T')$.
- Devise an FPT algorithm with respect to $d_{\text{SPR}}$.
- Sketch an approximation algorithm.
Kernelization – Subtrees

**Common subtree reduction.**

- Replace any subtree (with \( \geq 2 \) leaves) that occurs identically in both trees by a single leaf with a new label.
Kernelization – Subtrees

**Common subtree reduction.**
- Replace any subtree (with \( \geq 2 \) leaves) that occurs identically in both trees by a single leaf with a new label.
Kernelization – Subtrees

Common subtree reduction.
- Replace any subtree (with $\geq 2$ leaves) that occurs identically in both trees by a single leaf with a new label.
Common subtree reduction.

- Replace any subtree (with $\geq 2$ leaves) that occurs identically in both trees by a single leaf with a new label.

Lemma 5. Applying the common subtree reduction is safe, i.e., $d_{SPR}(T, T') = d_{SPR}(S, S')$. 
Kernelization – Subtrees

Common subtree reduction.
- Replace any subtree (with $\geq 2$ leaves) that occurs identically in both trees by a single leaf with a new label.

Suppose $S$ is covered by two trees of MAF.

**Lemma 5.** Applying the common subtree reduction is safe, i.e., $d_{SPR}(T, T') = d_{SPR}(S, S')$.

**Proof.**
Suppose is covered by two trees of MAF.
Kernelization – Subtrees

Common subtree reduction.
- Replace any subtree (with \( \geq 2 \) leaves) that occurs identically in both trees by a single leaf with a new label.

Lemma 5. Applying the common subtree reduction is safe, i.e., \( d_{SPR}(T, T') = d_{SPR}(S, S') \).

Proof.
Suppose is covered by two trees of MAF then there is an alternative MAF of the same size
Kernelization – Chains

**Chain reduction.**

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.
Kernelization – Chains

**Chain reduction.**
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.
Kernelization – Chains

Chain reduction.
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

\[ T \rightarrow T' \rightarrow S \rightarrow S' \]
Kernelization – Chains

**Chain reduction.**
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

\[
\text{Lemma 6. Applying chain reduction is safe, i.e., } d_{\text{SPR}}(T, T') = d_{\text{SPR}}(S, S').
\]
Kernelization – Chains

Chain reduction.
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

Lemma 6. Applying chain reduction is safe, i.e., $d_{\text{SPR}}(T, T') = d_{\text{SPR}}(S, S')$.

Proof.
- Show there is a tree with abc-chain in a MAF of $S$ and $S'$.
- Swap abc-chain with original chain for MAF of $T$ and $T'$. 
Kernelization – Chains

Chain reduction.

■ Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

Lemma 6. Applying chain reduction is safe, i.e., \(d_{\text{SPR}}(T, T') = d_{\text{SPR}}(S, S')\).

Proof.

■ Consider embedding of a MAF \(F\) into \(S\).
Kernelization – Chains

Chain reduction.
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

Lemma 6. Applying chain reduction is safe, i.e., \( d_{\text{SPR}}(T, T') = d_{\text{SPR}}(S, S') \).

Proof. Case 1
- Consider embedding of a MAF \( F \) into \( S \).
Kernelization – Chains

Chain reduction.
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

Lemma 6. Applying chain reduction is safe, i.e., \( d_{SPR}(T, T') = d_{SPR}(S, S') \).

Proof.  
- Consider embedding of a MAF \( F \) into \( S \).
Kernelization – Chains

Chain reduction.
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

\[
\begin{align*}
T & \quad T' \\
S & \quad S'
\end{align*}
\]

**Lemma 6.** Applying chain reduction is safe, i.e., \(d_{\text{SPR}}(T, T') = d_{\text{SPR}}(S, S')\).

**Proof.**
- Consider embedding of a MAF \(F\) into \(S\).

**Case 1**
Kernelization – Chains

**Chain reduction.**
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

- Consider embedding of a MAF $F$ into $S$.

**Lemma 6.** Applying chain reduction is safe, i.e., $d_{SPR}(T, T') = d_{SPR}(S, S')$.

**Proof.**
- Consider embedding of a MAF $F$ into $S$. 

**Case 2**
Kernelization – Chains

Chain reduction.

- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

\[ T \rightarrow T' \rightarrow S \rightarrow S' \]

Lemma 6. Applying chain reduction is safe, i.e., \( d_{\text{SPR}}(T, T') = d_{\text{SPR}}(S, S') \).

Proof.

- Consider embedding of a MAF \( F \) into \( S \).

Case 2
Kernelization – Chains

Chain reduction.
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

\[ T \rightarrow T' \rightarrow S \rightarrow S' \]

Lemma 6. Applying chain reduction is safe, i.e., \( d_{\text{SPR}}(T, T') = d_{\text{SPR}}(S, S') \).

Proof.
- Consider embedding of a MAF \( F \) into \( S \).
Kernelization – Chains

**Chain reduction.**
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

\[ T \quad T' \quad S \quad S' \]

**Lemma 6.** Applying chain reduction is safe, i.e., \( d_{SPR}(T, T') = d_{SPR}(S, S') \).

**Proof.**
- Consider embedding of a MAF \( F \) into \( S \).

**Case 3**
Kernelization – Chains

**Chain reduction.**
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

\[
\begin{array}{c}
T \\
\end{array} \quad \begin{array}{c}
T' \\
\end{array} \quad \begin{array}{c}
S \\
\end{array} \quad \begin{array}{c}
S' \\
\end{array}
\]

Why not using a chain of length \( \leq 2 \)?

**Lemma 6.** Applying chain reduction is safe, i.e., \( d_{SPR}(T, T') = d_{SPR}(S, S') \).
Kernelization – Chains

Chain reduction.
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

![Diagram showing chain reduction](image.png)

Why not using a chain of length $\leq 2$?

Lemma 6. Applying chain reduction is safe, i.e., $d_{\text{SPR}}(T, T') = d_{\text{SPR}}(S, S')$. 

![Diagram showing Lemma 6](image.png)
Kernelization – Chains

Chain reduction.
- Replace any chain of leaves that occurs identically (from bottom to top) in both trees by three new leaves.

\[ T \rightarrow T' \]
\[ S \rightarrow S' \]

Why not using a chain of length \( \leq 2 \)?

Lemma 6. Applying chain reduction is safe, i.e., \( d_{\text{SPR}}(T, T') = d_{\text{SPR}}(S, S') \).
Lemma 7. Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then

$$|X'| \leq 28 \cdot d_{SPR}(T, T').$$
Lemma 7.
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then

$$|X'| \leq 28 \text{d}_{\text{SPR}}(T, T').$$

Proof. Let $F = \{T_ρ, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. 

Kernel Size
Lemma 7.
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then

$$|X'| \leq 28 \cdot d_{SPR}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. 

Proof.
Kernel Size

**Lemma 7.** Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then

$$|X'| \leq 28 d_{\text{SPR}}(T, T').$$

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$. 

Proof.

Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$. 

Proof.

Kernel Size

**Lemma 7.** Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
$$|X'| \leq 28 \, d_{SPR}(T, T').$$

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

**Claim 1.** \( \sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k. \)

We know \( k = d_{SPR}(S, S') = d_{SPR}(T, T'). \)
Kernel Size

**Lemma 7.** Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then

$$|X'| \leq 28 \text{d}_{\text{SPR}}(T, T').$$

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$.

We know $k = \text{d}_{\text{SPR}}(S, S') = \text{d}_{\text{SPR}}(T, T')$. 

![Diagram](image-url)
Kernel Size

**Lemma 7.** Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
\[ |X'| \leq 28 \text{d}_{\text{SPR}}(T, T'). \]

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. \[ \sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k. \]

We know \[ k = d_{\text{SPR}}(S, S') = d_{\text{SPR}}(T, T'). \]
Kernel Size

**Lemma 7.** Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
\[ |X'| \leq 28 \, d_{SPR}(T, T'). \]

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$. We know $k = d_{SPR}(S, S') = d_{SPR}(T, T')$.

Claim 1. $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k.$

We know $k = d_{SPR}(S, S') = d_{SPR}(T, T')$. 

\[ |V(H)| = k + 1 \]
Kernel Size

**Lemma 7.** Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then

$$|X'| \leq 28 \text{d}_{\text{SPR}}(T, T').$$

**Proof.** Let $F = \{T_{\rho}, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

**Claim 1.** $\sum_{i=\rho}^k (n(T_i) + n'(T_i)) \leq 4k.$

We know

$k = d_{\text{SPR}}(S, S') = d_{\text{SPR}}(T, T').$

$$|V(H)| = k + 1 = |E(H)| + 1$$
Kernel Size

**Lemma 7.**
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
\[ |X'| \leq 28 \text{d}_{\text{SPR}}(T, T'). \]

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$. We know $k = \text{d}_{\text{SPR}}(S, S') = \text{d}_{\text{SPR}}(T, T')$.

\[ |V(H)| = k + 1 = |E(H)| + 1 \]
\[ \sum_{i=\rho}^{k} n(T_i) = 2|E(H)| \leq 2k \]
**Lemma 7.**
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
\[ |X'| \leq 28 \cdot d_{\text{SPR}}(T, T'). \]

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

**Claim 1.** \[ \sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k. \]

**Claim 2.** # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.

We know $k = d_{\text{SPR}}(S, S') = d_{\text{SPR}}(T, T')$. 

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $S$ and $S'$ be on $X'$. Then
\[ |X'| \leq 28 \cdot d_{\text{SPR}}(T, T'). \]
Kernel Size

**Lemma 7.**
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then

$$|X'| \leq 28 d_{\text{SPR}}(T, T').$$

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

**Claim 1.** $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$.

**Claim 2.** $\# \text{ leaves of } T_i \leq 7(n(T_i) + n'(T_i))$.

We know $k = d_{\text{SPR}}(S, S') = d_{\text{SPR}}(T, T')$. 

Proof. 

Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

We know $k = d_{\text{SPR}}(S, S') = d_{\text{SPR}}(T, T')$. 

$T_i$

\[
\begin{array}{c}
\text{root} \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \\
/ \
\end{array}
\]
Kernel Size

Lemma 7.
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then

$$|X'| \leq 28 d_{\text{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$.

Claim 2. $\# \text{ leaves of } T_i \leq 7(n(T_i) + n'(T_i))$.

We know $k = d_{\text{SPR}}(S, S') = d_{\text{SPR}}(T, T')$. 

\[\text{Proof.}\] Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $S$ and $S'$ be on $X'$. Then

$$|X'| \leq 28 d_{\text{SPR}}(T, T').$$

Lemma 7.
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then

$$|X'| \leq 28 d_{\text{SPR}}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$.

Claim 2. $\# \text{ leaves of } T_i \leq 7(n(T_i) + n'(T_i))$.

We know $k = d_{\text{SPR}}(S, S') = d_{\text{SPR}}(T, T')$. 

\[\text{Proof.}\] 

\[\text{We know}\] 

\[k = d_{\text{SPR}}(S, S') = d_{\text{SPR}}(T, T').\]
Kernel Size

**Lemma 7.**
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
\[ |X'| \leq 28 \, d_{SPR}(T, T'). \]

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$. We know
\[ k = d_{SPR}(S, S') = d_{SPR}(T, T'). \]

**Claim 1.** $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k.$

**Claim 2.** # leaves of $T_i \leq 7(n(T_i) + n'(T_i)).$
Kernel Size

**Lemma 7.** Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then $|X'| \leq 28 \text{d}_{\text{SPR}}(T, T')$.

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.

We know $k = d_{\text{SPR}}(S, S') = d_{\text{SPR}}(T, T')$. 

**Proof.** Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

We know $k = d_{\text{SPR}}(S, S') = d_{\text{SPR}}(T, T')$. 

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $S$ and $S'$ be on $X'$. Then $|X'| \leq 28 \text{d}_{\text{SPR}}(T, T')$. 

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$.

Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$. 

We know $k = d_{\text{SPR}}(S, S') = d_{\text{SPR}}(T, T')$. 

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$.

Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$. 

We know $k = d_{\text{SPR}}(S, S') = d_{\text{SPR}}(T, T')$. 

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$.

Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.
Lemma 7.
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
$$|X'| \leq 28 \text{spr}(T, T').$$

Proof. Let $F = \{T_{\rho}, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$. 

We know $k = \text{spr}(S, S') = \text{spr}(T, T')$. 

Proof. 

\begin{align*}
\text{Similarly, let } n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|. \\
\text{We know } k = \text{spr}(S, S') = \text{spr}(T, T').
\end{align*}
Kernel Size

Lemma 7.
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then

$$|X'| \leq 28 \cdot d_{SPR}(T, T').$$

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.  

We know $k = d_{SPR}(S, S') = d_{SPR}(T, T')$. 
Kernel Size

Lemma 7.
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
\[ |X'| \leq 28 \text{d}_{\text{SPR}}(T, T'). \]

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Proof. Let $F = \{T_\rho, T_1, ..., T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j | T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. \(\sum_{i=\rho}^{k}(n(T_i) + n'(T_i)) \leq 4k.\)

Claim 2. \# leaves of $T_i \leq 7(n(T_i) + n'(T_i)).$
Kernel Size

**Lemma 7.**
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then

$$|X'| \leq 28 \, d_{SPR}(T, T').$$

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

**Claim 1.** $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$.

**Claim 2.** $\#$ leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.

We know $k = d_{SPR}(S, S') = d_{SPR}(T, T')$. 

Proof.

Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$. 

We know $k = d_{SPR}(S, S') = d_{SPR}(T, T')$. 

Proof.

| $X'| = \sum_{i=\rho}^{k} \#$ leaves of $T_i$
Kernel Size

Lemma 7.
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then
\[ |X'| \leq 28 \cdot d_{SPR}(T, T'). \]

Proof. Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

Claim 1. $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$.

Claim 2. # leaves of $T_i \leq 7(n(T_i) + n'(T_i))$.

We know $k = d_{SPR}(S, S') = d_{SPR}(T, T')$. 

<table>
<thead>
<tr>
<th>$X'$</th>
<th>$\leq \sum_{i=\rho}^{k}$ # leaves of $T_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\leq \sum_{i=\rho}^{k} 7(n(T_i) + n'(T_i))$</td>
<td>$\leq 28 \cdot d_{SPR}(T, T')$</td>
</tr>
</tbody>
</table>
 Lemma 7.
Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules. Let $S$ and $S'$ be on $X'$. Then

$$|X'| \leq 28 d_{\text{SPR}}(T, T').$$

**Proof.** Let $F = \{T_\rho, T_1, \ldots, T_k\}$ be MAF for $S$ and $S'$. Let $n(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S\}|$. Similarly, let $n'(T_i) := |\{T_j \mid T_j \in F \land T_i \text{ and } T_j \text{ touch in } S'\}|$.

**Claim 1.** $\sum_{i=\rho}^{k} (n(T_i) + n'(T_i)) \leq 4k$.

**Claim 2.** The number of leaves of $T_i \leq 7(n(T_i) + n'(T_i))$. Then

$$|X'| = \sum_{i=\rho}^{k} \# \text{ leaves of } T_i \leq \sum_{i=\rho}^{k} 7(n(T_i) + n'(T_i)) \leq 28k.$$
Theorem 8.
Computing $d_{SPR}(T, T')$ is fixed-parameter tractable when parameterized by $d_{SPR}(T, T')$. 
FPT Algorithm

Theorem 8.
Computing $d_{SPR}(T, T')$ is fixed-parameter tractable when parameterized by $d_{SPR}(T, T')$.

Proof.

- Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules.
- Let $S$ and $S'$ be on $X'$ and let $k = d_{SPR}(S, S')$. 

Theorem 8.
Computing $d_{SPR}(T, T')$ is fixed-parameter tractable when parameterized by $d_{SPR}(T, T')$.

Proof.

- Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules.
- Let $S$ and $S'$ be on $X'$ and let $k = d_{SPR}(S, S')$. 

FPT Algorithm

Theorem 8.
Computing $d_{\text{SPR}}(T, T')$ is fixed-parameter tractable when parameterized by $d_{\text{SPR}}(T, T')$.

Proof.
- Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules.
- Let $S$ and $S'$ be on $X'$ and let $k = d_{\text{SPR}}(S, S')$.
- $S$ has at most $4|X'|^2$ neighbors in the SPR-graph $G$. 
FPT Algorithm

Theorem 8.
Computing $d_{SPR}(T, T')$ is fixed-parameter tractable when parameterized by $d_{SPR}(T, T')$.

Proof.
- Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules.
- Let $S$ and $S'$ be on $X'$ and let $k = d_{SPR}(S, S')$.

- $S$ has at most $4|X'|^2$ neighbors in the SPR-graph $G$.
  - $S$ has less than $2|X'|$ edges to cut and to attach to.
**FPT Algorithm**

**Theorem 8.**
Computing $d_{\text{SPR}}(T, T')$ is fixed-parameter tractable when parameterized by $d_{\text{SPR}}(T, T')$.

**Proof.**
- Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules.
- Let $S$ and $S'$ be on $X'$ and let $k = d_{\text{SPR}}(S, S')$.
- $S$ has at most $4|X'|^2$ neighbors in the SPR-graph $G$.
  - $S$ has less than $2|X'|$ edges to cut and to attach to.
- Length-$k$ BFS from $S$ visits at most $O\left((4|X'|^2)^k\right) = O\left((56k)^{2k}\right)$ trees.
FPT Algorithm

**Theorem 8.** Computing $d_{SPR}(T, T')$ is fixed-parameter tractable when parameterized by $d_{SPR}(T, T')$.

**Proof.**

- Reduce $T$ and $T'$ to $S$ and $S'$ by exhaustively applying the reduction rules.
- Let $S$ and $S'$ be on $X'$ and let $k = d_{SPR}(S, S')$.

- $S$ has at most $4|X'|^2$ neighbors in the SPR-graph $G$.
  - $S$ has less than $2|X'|$ edges to cut and to attach to. by Lemma 7

- Length-$k$ BFS from $S$ visits at most $O\left((4|X'|^2)^k\right) = O\left((56k)^{2k}\right)$ trees.
FPT Algorithm

Theorem 8.
Computing \( d_{\text{SPR}}(T, T') \) is fixed-parameter tractable when parameterized by \( d_{\text{SPR}}(T, T') \).

Proof.
- Reduce \( T \) and \( T' \) to \( S \) and \( S' \) by exhaustively applying the reduction rules.
- Let \( S \) and \( S' \) be on \( X' \) and let \( k = d_{\text{SPR}}(S, S') \).
- \( S \) has at most \( 4|X'|^2 \) neighbors in the SPR-graph \( G \).
  - \( S \) has less than \( 2|X'| \) edges to cut and to attach to.
- Length-\( k \) BFS from \( S \) visits at most \( O\left(\left(4|X'|^2\right)^k\right) = O\left((56k)^{2k}\right) \) trees.
- Since \( k = d_{\text{SPR}}(S, S') = d_{\text{SPR}}(T, T') \), this yields an FPT algorithm.
Approximation Algorithm

Idea.
- Given trees $T$ and $T'$, which are reduced by the previous rules, we compute an agreement forest $F$ by
  - successively making “cuts” and “eliminations”.
  - These steps let $T$ and $T'$ shrink further and further.
- Show that $|F|$ is at most $3|F^*|$, where $F^*$ is a MAF of $T$ and $T'$. 
Approximation Algorithm

\textsc{approxDSPR}(T, T')

\begin{align*}
i & \leftarrow 1 \\
G_i & \leftarrow T \\
H_i & \leftarrow T'
\end{align*}

while \exists \text{ pair of sibling leaves } a \text{ and } b \text{ in } G_i \text{ do}

return $|H_i| - 1$
Approximation Algorithm

\text{APPROXDSPR}(T, T')

\begin{align*}
i &\leftarrow 1 \\
G_i &\leftarrow T \\
H_i &\leftarrow T' \\
\text{while } &\exists \text{ pair of sibling leaves } a \text{ and } b \text{ in } G_i \text{ do} \\
\text{find the case that applies to } a \text{ and } b \text{ in } H_i &\\
\text{apply the corresponding modification} &\\
\text{to obtain } G_i+1 &\text{ from } G_i \text{ and } H_i+1 &\text{ from } H_i \\
i &+ + \\
\text{return } &|H_i| - 1
\end{align*}
Approximation Algorithm

\[ \text{APPROXDSPR}(T, T') \]

\[
i \leftarrow 1
\]
\[
G_i \leftarrow T
\]
\[
H_i \leftarrow T'
\]

\[\text{while } \exists \text{ pair of sibling leaves } a \text{ and } b \text{ in } G_i \text{ do} \]

\[\text{find the case that applies to } a \text{ and } b \text{ in } H_i \]

\[\text{apply the corresponding modification to obtain } G_{i+1} \text{ from } G_i \text{ and } H_{i+1} \text{ from } H_i \]

\[i \leftarrow i + 1\]

\[\text{return } |H_i| - 1\]
Approximation Algorithm

\texttt{APPROXDSPR}(T, T')

\begin{align*}
i & \leftarrow 1 \\
G_i & \leftarrow T \\
H_i & \leftarrow T'
\end{align*}

\textbf{while} \; \exists \; \text{pair of sibling leaves } a \text{ and } b \text{ in } G_i \; \textbf{do}

\begin{align*}
& \text{find the case that applies to } a \text{ and } b \text{ in } H_i \\
& \text{apply the corresponding modification to obtain } G_i+1 \text{ from } G_i \text{ and } H_i+1 \text{ from } H_i
\end{align*}

\textbf{return} \; |H_i| - 1  

\texttt{Case 1}

\begin{align*}
G_i & \\
H_i &
\end{align*}
Approximation Algorithm

\texttt{APPROXDSPR}(T, T')

\begin{align*}
i & \leftarrow 1 \\
G_i & \leftarrow T \\
H_i & \leftarrow T'
\end{align*}

\textbf{while} \ \exists \ \text{pair of sibling leaves } a \text{ and } b \text{ in } G_i \textbf{ do}

\begin{align*}
\text{find the case that applies to } a \text{ and } b \text{ in } H_i \\
\text{apply the corresponding modification to obtain } G_{i+1} \text{ from } G_i \text{ and } H_{i+1} \text{ from } H_i
\end{align*}

\textbf{return} |H_i| - 1
Approximation Algorithm

\textsc{approxDSPR}(T, T')

\begin{align*}
i &\leftarrow 1 \\
G_i &\leftarrow T \\
H_i &\leftarrow T'
\end{align*}

while \exists \text{ pair of sibling leaves } a \text{ and } b \text{ in } G_i \text{ do}

\begin{align*}
&\quad \text{find the case that applies to } a \text{ and } b \text{ in } H_i \\
&\quad \text{apply the corresponding modification}
\end{align*}

\text{return } |H_i| - 1
Approximation Algorithm

\textit{APPROXDSPR}(T, T')

\begin{align*}
i & \leftarrow 1 \\
G_i & \leftarrow T \\
H_i & \leftarrow T'
\end{align*}

\textbf{while} \exists \text{ pair of sibling leaves } a \text{ and } b \text{ in } G_i \text{ do}

\begin{align*}
\text{find the case that applies to } a \text{ and } b \text{ in } H_i \\
\text{apply the corresponding modification}
\end{align*}

\textbf{return } |H_i| - 1
Approximation Algorithm

\text{APPROXDSPR}(T, T')

\begin{align*}
i &\leftarrow 1 \\
G_i &\leftarrow T \\
H_i &\leftarrow T'
\end{align*}

\textbf{while} \ \exists \ \text{pair of sibling leaves} \ a \ \text{and} \ b \ \text{in} \ G_i \ \textbf{do}

\begin{align*}
&\text{find the case that applies to} \ a \ \text{and} \ b \ \text{in} \ H_i \\
&\text{apply the corresponding modification} \\
&\text{to obtain} \ G_{i+1} \ \text{from} \ G_i \ \text{and} \ H_{i+1} \ \text{from} \ H_i \\
&i + +
\end{align*}

\textbf{return} \ |H_i| - 1

\begin{align*}
\text{Case 1} && \text{Case 2} && \text{Case 3} && \text{Case 4}
\end{align*}
Approximation Algorithm – Example

\[ T = G_1 \]

\[ T' = H_1 \]
Approximation Algorithm – Example

$T = G_1$

$T' = H_1$
Approximation Algorithm – Example

\[ T = G_1 \]

\[ T' = H_1 \]

Should we cut off leaf 1 or leaf 2 or everything between them in \( H_1 \)?
Approximation Algorithm – Example

$T = G_1$

$T' = H_1$

**Case 2**

- Should we cut off leaf 1 or leaf 2 or everything between them in $H_1$?
- Do parts of each!
Approximation Algorithm – Example

Case 2

- Should we cut off leaf 1 or leaf 2 or everything between them in $H_1$?
- Do parts of each!

Diagram:

- $G_2$
  - $\rho$
  - 3, 4, 5, 6

- $H_2$
  - $\rho$
  - 3, 4, 6
Approximation Algorithm – Example

\[ G_2 \]

\[ H_2 \]
Case 1

- If the same "cherry" (i.e., pair of leaves) occurs in $G_i$ and $H_i$, we simply reduce it.
Approximation Algorithm – Example

Case 1

If the same “cherry” (i.e., pair of leaves) occurs in $G_i$ and $H_i$, we simply reduce it.
Approximation Algorithm – Example

Case 4
- Leaf $b$ is the only leaf of a tree in $H_i$.
- Cut off $b$ in $G_i$. 
Approximation Algorithm – Example

Return 3.
Approximation algorithm – analysis

Case  | $G_i$  | $H_i$  | $G_{i+1}$ | $H_{i+1}$ | Cost

1  | \(a\) \(b\)  | \(a\) \(b\)  | \(c\)  | \(c\)  |
Approximation algorithm – analysis

Case  | $G_i$  | $H_i$  | $G_{i+1}$  | $H_{i+1}$  | Cost
--- | --- | --- | --- | --- | ---
1  | \( \begin{array}{cc} a & b \\ \end{array} \) | \( \begin{array}{cc} a & b \\ \end{array} \) | \( \begin{array}{c} c \\ \end{array} \) | \( \begin{array}{c} c \\ \end{array} \) | no mistake
Approximation algorithm – analysis

Case  | $G_i$  | $H_i$  | $G_{i+1}$  | $H_{i+1}$  | Cost
--- | --- | --- | --- | --- | ---
1 | ![Tree](image1.png) | ![Tree](image2.png) | ![Tree](image3.png) | ![Tree](image4.png) | no mistake
2 | ![Tree](image5.png) | ![Tree](image6.png) | ![Tree](image7.png) | ![Tree](image8.png) |
Approximation algorithm – analysis

<table>
<thead>
<tr>
<th>Case</th>
<th>$G_i$</th>
<th>$H_i$</th>
<th>$G_{i+1}$</th>
<th>$H_{i+1}$</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a \ b$</td>
<td>$a \ b$</td>
<td>$c$</td>
<td>$c$</td>
<td>no mistake</td>
</tr>
<tr>
<td>2</td>
<td>$a \ b$</td>
<td>$a \ b$</td>
<td>$a \ b$</td>
<td>$a \ b$</td>
<td>3 cuts 1+ good</td>
</tr>
</tbody>
</table>
## Approximation algorithm – analysis

<table>
<thead>
<tr>
<th>Case</th>
<th>$G_i$</th>
<th>$H_i$</th>
<th>$G_{i+1}$</th>
<th>$H_{i+1}$</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
<td><img src="image3" alt="Diagram" /></td>
<td><img src="image4" alt="Diagram" /></td>
<td>no mistake</td>
</tr>
<tr>
<td>2</td>
<td><img src="image5" alt="Diagram" /></td>
<td><img src="image6" alt="Diagram" /></td>
<td><img src="image7" alt="Diagram" /></td>
<td><img src="image8" alt="Diagram" /></td>
<td>3 cuts 1+ good</td>
</tr>
<tr>
<td>3</td>
<td><img src="image9" alt="Diagram" /></td>
<td><img src="image10" alt="Diagram" /></td>
<td><img src="image11" alt="Diagram" /></td>
<td><img src="image12" alt="Diagram" /></td>
<td></td>
</tr>
</tbody>
</table>
Approximation algorithm – analysis

<table>
<thead>
<tr>
<th>Case</th>
<th>$G_i$</th>
<th>$H_i$</th>
<th>$G_{i+1}$</th>
<th>$H_{i+1}$</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no mistake</td>
</tr>
<tr>
<td></td>
<td>$a$</td>
<td>$b$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$a$</td>
<td>$b$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 cuts</td>
</tr>
<tr>
<td></td>
<td>$a$</td>
<td>$b$</td>
<td></td>
<td></td>
<td>1+ good</td>
</tr>
<tr>
<td></td>
<td>$a$</td>
<td>$b$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$a$</td>
<td>$b$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 cuts</td>
</tr>
<tr>
<td></td>
<td>$a$</td>
<td>$b$</td>
<td></td>
<td></td>
<td>1+ good</td>
</tr>
<tr>
<td></td>
<td>$a$</td>
<td>$b$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$a$</td>
<td>$b$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Approximation algorithm – analysis

<table>
<thead>
<tr>
<th>Case</th>
<th>$G_i$</th>
<th>$H_i$</th>
<th>$G_{i+1}$</th>
<th>$H_{i+1}$</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a$</td>
<td>$b$</td>
<td></td>
<td>$c$</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>$a$</td>
<td>$b$</td>
<td>$a$</td>
<td>$b$</td>
<td>3 cuts</td>
</tr>
<tr>
<td></td>
<td>$a$</td>
<td>$b$</td>
<td>$a$</td>
<td>$b$</td>
<td>1+ good</td>
</tr>
<tr>
<td>3</td>
<td>$a$</td>
<td>$b$</td>
<td>$a$</td>
<td>$b$</td>
<td>2 cuts</td>
</tr>
<tr>
<td></td>
<td>$a$</td>
<td>$b$</td>
<td>$a$</td>
<td>$b$</td>
<td>1+ good</td>
</tr>
<tr>
<td>4</td>
<td>$a$</td>
<td>$b$</td>
<td>$a$</td>
<td>$b$</td>
<td></td>
</tr>
</tbody>
</table>
Approximation algorithm – analysis

<table>
<thead>
<tr>
<th>Case</th>
<th>$G_i$</th>
<th>$H_i$</th>
<th>$G_{i+1}$</th>
<th>$H_{i+1}$</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image1" alt="Tree 1" /></td>
<td><img src="image2" alt="Tree 2" /></td>
<td><img src="image3" alt="Tree 3" /></td>
<td><img src="image4" alt="Tree 4" /></td>
<td>no mistake</td>
</tr>
<tr>
<td>2</td>
<td><img src="image5" alt="Tree 5" /></td>
<td><img src="image6" alt="Tree 6" /></td>
<td><img src="image7" alt="Tree 7" /></td>
<td><img src="image8" alt="Tree 8" /></td>
<td>3 cuts 1+ good</td>
</tr>
<tr>
<td>3</td>
<td><img src="image9" alt="Tree 9" /></td>
<td><img src="image10" alt="Tree 10" /></td>
<td><img src="image11" alt="Tree 11" /></td>
<td><img src="image12" alt="Tree 12" /></td>
<td>2 cuts 1+ good</td>
</tr>
<tr>
<td>4</td>
<td><img src="image13" alt="Tree 13" /></td>
<td><img src="image14" alt="Tree 14" /></td>
<td><img src="image15" alt="Tree 15" /></td>
<td><img src="image16" alt="Tree 16" /></td>
<td>1 cut 1 good</td>
</tr>
</tbody>
</table>
Approximation algorithm – analysis

<table>
<thead>
<tr>
<th>Case</th>
<th>$G_i$</th>
<th>$H_i$</th>
<th>$G_{i+1}$</th>
<th>$H_{i+1}$</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 cuts</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 cuts</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 cut</td>
</tr>
</tbody>
</table>

Theorem 9

**APPROXDSPR** is a 3-approximation algorithm for $d_{SPR}(T, T')$ with an $O(|X|^2)$ running time.
Discussion

Kernelization.
- Kernelization is an important technique to construct FPT algorithms.
- Result important since SPR-distance small in practice.
- Reduction rules actually give a kernel of size at most $15k - 9$ (we have shown $28k$).
- With further reduction rules, we can get a size below $11k - 9$. [KL '18]
- Divide & conquer techniques can (in practice) further reduce the problem sizes. [LS '11]
Discussion

Kernelization.
- Kernelization is an important technique to construct FPT algorithms.
- Result important since SPR-distance small in practice.
- Reduction rules actually give a kernel of size at most $15k - 9$ (we have shown $28k$).
- With further reduction rules, we can get a size below $11k - 9$. [KL '18]
- Divide & conquer techniques can (in practice) further reduce the problem sizes. [LS '11]

Approximation algorithm.
- There exists a 2-approximation algorithms for the SPR-distance with a running time in $O(n^3)$. [CHW '17]
Discussion

**Phylogenetic trees.**

- There are other classes of phylogenetic trees: unrooted, non-binary, ranked, ... 
- Trees can be generalized to **phylogenetic networks**, which can also have indegree 2 outdegree 1 vertices.
Discussion

Phylogenetic trees.

- There are other classes of phylogenetic trees: unrooted, non-binary, ranked, ...

- Trees can be generalized to phylogenetic networks, which can also have indegree 2 outdegree 1 vertices.

Maximum Agreement Forests.

- Reframing (characterizing) a problem in a different way, can sometimes make your life a lot easier.

- MAF can be generalized to Maximum Agreement Graphs, but these do not characterize the SPR-distance of networks anymore. [K '20]
Original papers:
■ [BS ’05] Semple C., Bordewich M.: *On the computational complexity of the rooted subtree prune and regraft distance* (for SPR, MAF, characterisation, fpt, divide & conquer)
■ [RSW ’06] Rodrigues E. M., Sagot M.-F., Wakabayashi Y.: *The maximum agreement forest problem: Approximation algorithms and computational experiments* (for approx. algorithm)

Referenced papers:
■ [LS ’11] Linz S., Semple C.: *A cluster reduction for computing the subtree distance between phylogenies*