Faster Computation of the Robinson-Foulds Distance between Phylogenetic Networks

- Tetsuo Asano (JAIST, Japan)
- Jesper Jansson (Ochanomizu University, Japan)
- Kunihiko Sadakane (NII, Japan)
- Ryuhei Uehara (JAIST, Japan)
- Gabriel Valiente (Technical University of Catalonia, Spain)

Introduction

Definition

A phylogenetic tree is a rooted, unordered tree with distinctly labeled leaves.

Introduction

Definition

A phylogenetic tree is a rooted, unordered tree with distinctly labeled leaves.

Can describe divergent evolutionary history for a set of objects, where:

“objects” = Biological species, categories of species, populations, proteins, nucleic acids, natural languages, chain letters, medieval manuscripts, or ...

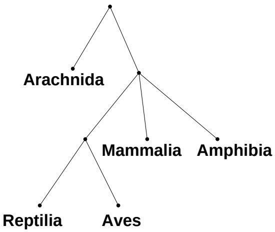
Introduction

Definition

A phylogenetic tree is a rooted, unordered tree with distinctly labeled leaves.

Can describe divergent evolutionary history for a set of objects, where:

“objects” = Biological species, categories of species, populations, proteins, nucleic acids, natural languages, chain letters, medieval manuscripts, or ...



Main idea:

- Represent objects by *leaves* in the tree.
- Select branching structure so that *internal nodes* correspond to common ancestors.

Phylogenetic networks

Sometimes, the objects fail to fit the phylogenetic tree model.

Phylogenetic networks

Sometimes, the objects fail to fit the phylogenetic tree model.

- Horizontal gene transfer
- Hybrid speciation

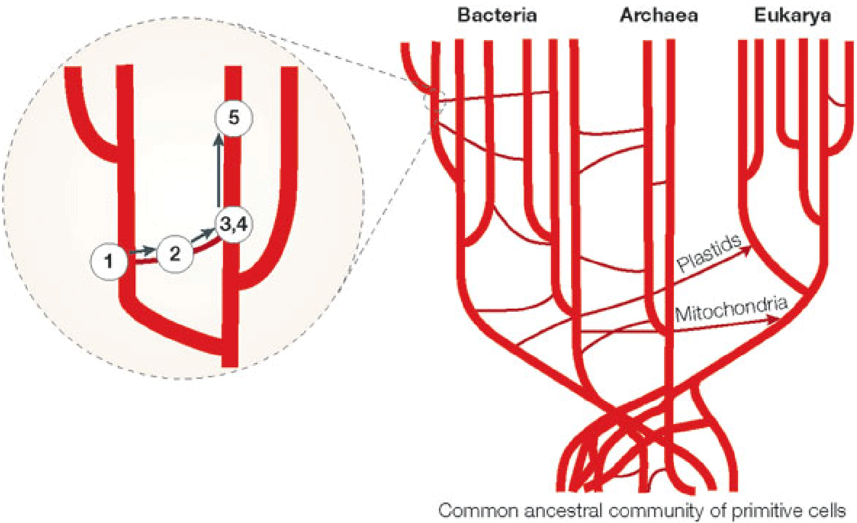
Phylogenetic networks

Sometimes, the objects fail to fit the phylogenetic tree model.

- Horizontal gene transfer
- Hybrid speciation

Phylogenetic network: Generalization of rooted phylogenetic tree in which internal nodes may have more than one parent.

Phylogenetic network, example 1



(From Smets & Barkay, *Nature Reviews Microbiology*, Vol. 3, pp. 675–678, 2005).

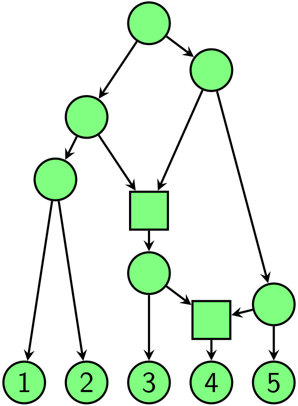
Definition of phylogenetic network

Definition

A phylogenetic network is a connected, rooted, simple directed acyclic graph in which:

- Nodes with indegree ≤ 1 are called *tree nodes* and nodes with indegree ≥ 2 are called *hybrid nodes*.
- No node has both indegree 1 and outdegree 1.
- All nodes with outdegree 0 are distinctly labeled ("leaves").

Phylogenetic network, example 2



Robinson-Foulds distance

Let $N = (V, E)$ be a given phylogenetic network.

- For any nodes $u, v \in V$, v is a *descendant of* u if v is reachable from u in N. (For convenience, v is a descendant of itself.)

Robinson-Foulds distance

Let $N = (V, E)$ be a given phylogenetic network.

- For any nodes $u, v \in V$, v is a *descendant of u* if v is reachable from u in N. (For convenience, v is a descendant of itself.)
- For any $v \in V$, define *the cluster of v* (denoted by $C(v)$) as the set of all leaves which are descendants of v.

Robinson-Foulds distance

Let $N = (V, E)$ be a given phylogenetic network.

- For any nodes $u, v \in V$, v is a *descendant of u* if v is reachable from u in N. (For convenience, v is a descendant of itself.)

- For any $v \in V$, define *the cluster of v* (denoted by $C(v)$) as the set of all leaves which are descendants of v.

- *The cluster collection of N* is the multiset $C(N) = \left\{C(v) : v \in V\right\}$.

Robinson-Foulds distance

Let $N = (V, E)$ be a given phylogenetic network.

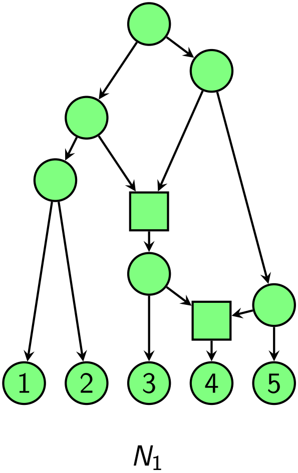
- For any nodes $u, v \in V$, v is a *descendant of u* if v is reachable from u in N. (For convenience, v is a descendant of itself.)
- For any $v \in V$, define *the cluster of v* (denoted by $C(v)$) as the set of all leaves which are descendants of v.
- *The cluster collection of N* is the multiset $C(N) = \{C(v) : v \in V\}$.

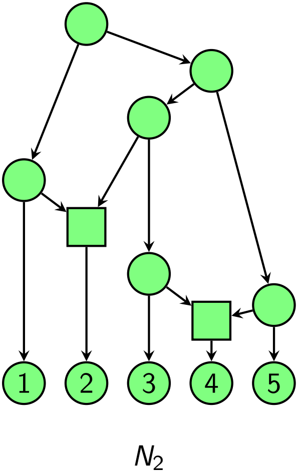
Definition

The Robinson-Foulds distance between two phylogenetic networks N_1, N_2 is:

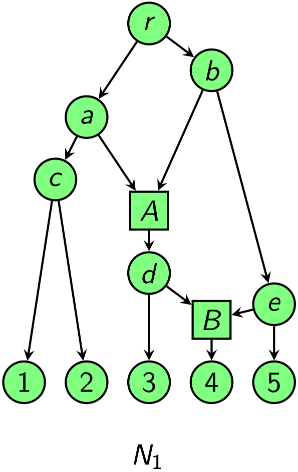
$$d_{RF}(N_1, N_2) = \frac{|C(N_1) \setminus C(N_2)| + |C(N_2) \setminus C(N_1)|}{2}$$

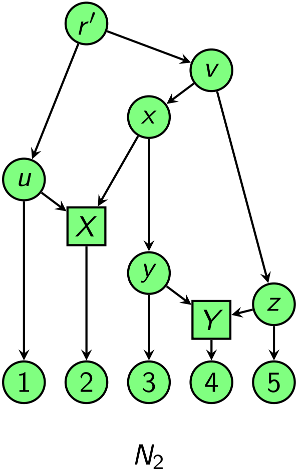
Robinson-Foulds distance, example



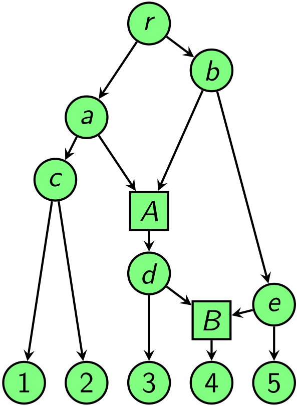


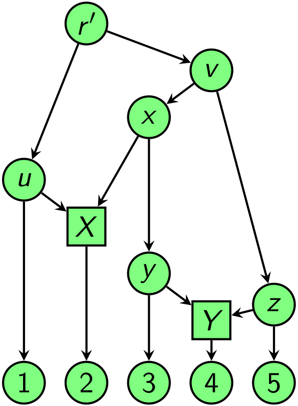
Robinson-Foulds distance, example





Robinson-Foulds distance, example

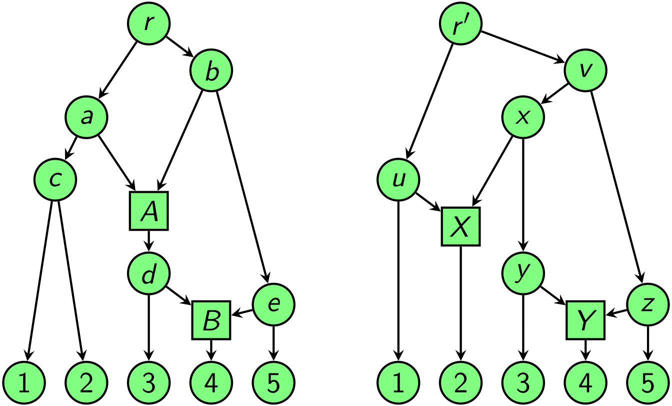




r	$\{1, 2, 3, 4, 5\}$	b	$\{3, 4, 5\}$
a	$\{1, 2, 3, 4\}$	A	$\{3, 4\}$
c	$\{1, 2\}$	d	$\{3, 4\}$
e	$\{4, 5\}$	B	$\{4\}$

r'	$\{1, 2, 3, 4, 5\}$	v	$\{2, 3, 4, 5\}$
u	$\{1, 2\}$	x	$\{2, 3, 4\}$
y	$\{3, 4\}$	z	$\{4, 5\}$
X	$\{2\}$	Y	$\{4\}$

Robinson-Foulds distance, example

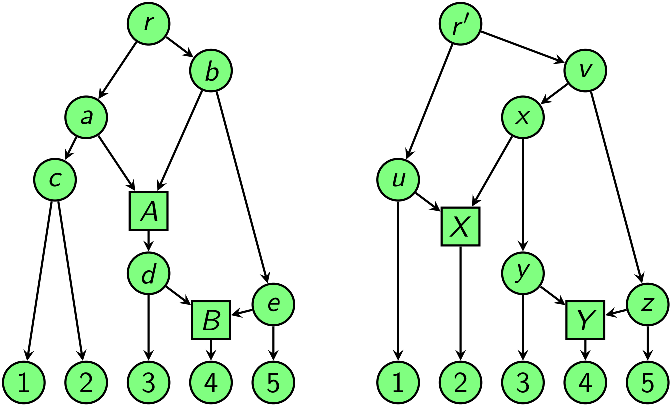


r	$\{1,2,3,4,5\}$	b	$\{3,4,5\}$
a	$\{1,2,3,4\}$	A	$\{3,4\}$
c	$\{1,2\}$	d	$\{3,4\}$
e	$\{4,5\}$	B	$\{4\}$

r'	$\{1,2,3,4,5\}$	v	$\{2,3,4,5\}$
u	$\{1,2\}$	x	$\{2,3,4\}$
y	$\{3,4\}$	z	$\{4,5\}$
X	$\{2\}$	Y	$\{4\}$

$C(r) = \{1,2,3,4,5\}$, $C(b) = \{3,4,5\}$, etc. and $C(N_1) = \{\{1,2,3,4,5\},\{3,4,5\},\ldots\}$.

Robinson-Foulds distance, example

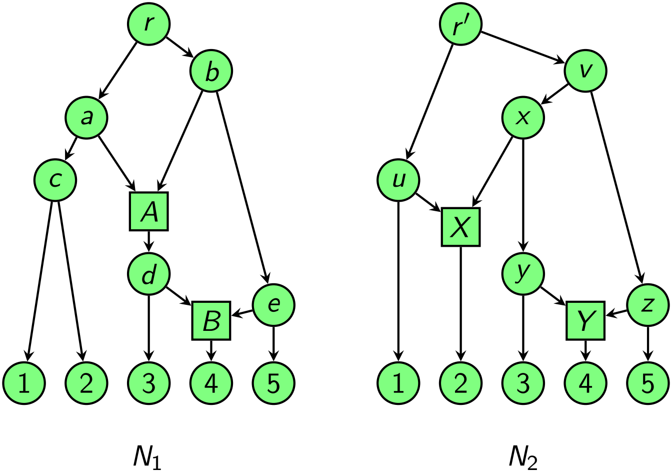


N_1 N_2

$$C(N_1) = \Big\{\{1,2,3,4,5\}, \{3,4,5\}, \{1,2,3,4\}, \{3,4\}, \{1,2\}, \{3,4\}, \{4,5\}, \{4\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\Big\}$$

$$C(N_2) = \Big\{\{1,2,3,4,5\}, \{2,3,4,5\}, \{1,2\}, \{2,3,4\}, \{3,4\}, \{4,5\}, \{2\}, \{4\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\Big\}$$

Robinson-Foulds distance, example



$C(N_1) = \Big\{\{1,2,3,4,5\}, \{3,4,5\}, \{1,2,3,4\}, \{3,4\}, \{1,2\}, \{3,4\}, \{4,5\}, \{4\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\Big\}$

$C(N_2) = \Big\{\{1,2,3,4,5\}, \{2,3,4,5\}, \{1,2\}, \{2,3,4\}, \{3,4\}, \{4,5\}, \{2\}, \{4\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\Big\}$

This gives $d_{RF}(N_1, N_2) = \frac{|C(N_1) \setminus C(N_2)| + |C(N_2) \setminus C(N_1)|}{2} = 3.$

Robinson-Foulds distance, cont.

The Robinson-Foulds distance $d_{RF}(N_1, N_2)$ measures the number of clusters that are not shared by N_1 and N_2. $\Rightarrow$ Measures their *dissimilarity*.

Robinson-Foulds distance, cont.

The Robinson-Foulds distance $d_{RF}(N_1, N_2)$ measures the number of clusters that are not shared by N_1 and N_2. $\Rightarrow$ Measures their *dissimilarity*.

Useful when comparing phylogenetic networks produced by alternative methods (or the same method applied to different data sets).

Robinson-Foulds distance, cont.

The Robinson-Foulds distance $d_{RF}(N_1, N_2)$ measures the number of clusters that are not shared by N_1 and N_2. $\Rightarrow$ Measures their *dissimilarity*.

Useful when comparing phylogenetic networks produced by alternative methods (or the same method applied to different data sets).

Remark 1:
d_{RF} is a metric on many biologically meaningful classes of phylogenetic networks, such as the so-called *regular phylogenetic networks*.
(Not a metric on arbitrary phylogenetic networks, though!)

Robinson-Foulds distance, cont.

The Robinson-Foulds distance $d_{RF}(N_1, N_2)$ measures the number of clusters that are not shared by N_1 and N_2. $\Rightarrow$ Measures their *dissimilarity*.

Useful when comparing phylogenetic networks produced by alternative methods (or the same method applied to different data sets).

Remark 1:
d_{RF} is a metric on many biologically meaningful classes of phylogenetic networks, such as the so-called *regular phylogenetic networks*.
(Not a metric on arbitrary phylogenetic networks, though!)

Remark 2: Other distances have been proposed in the literature:

- Tripartition distance [Moret *et al.*]: Further divide the descendant leaves into strict and non-strict descendants.
- Path-multiplicity distance (μ-distance) [Valiente *et al.*]: Take into account the number of paths from every node to each leaf.

Computing d_{RF}

Time complexity of computing $d_{RF}(N_1, N_2)$, where N_1 and N_2 contain n leaves, m nodes, and e edges in total:

- Simple method to compute $C(N)$ used by Cardona *et al.* [2009]: Breadth-first search from each node v to find the cluster $C(v)$.
  $\Rightarrow O(m\,e)$ time and $O(n\,m)$ space.

Computing d_{RF}

Time complexity of computing $d_{RF}(N_1, N_2)$, where N_1 and N_2 contain n leaves, m nodes, and e edges in total:

- Simple method to compute $C(N)$ used by Cardona *et al.* [2009]: Breadth-first search from each node v to find the cluster $C(v)$.
  $\Rightarrow O(m\,e)$ time and $O(n\,m)$ space.
- Special case where N_1, N_2 are phylogenetic *trees*: A classic algorithm by Day [1985] solves the problem in $O(n)$ time and $O(n)$ space.

Computing d_{RF}

Time complexity of computing $d_{RF}(N_1, N_2)$, where N_1 and N_2 contain n leaves, m nodes, and e edges in total:

- Simple method to compute $C(N)$ used by Cardona *et al.* [2009]: Breadth-first search from each node v to find the cluster $C(v)$.
  $\Rightarrow O(m\,e)$ time and $O(n\,m)$ space.
- Special case where N_1, N_2 are phylogenetic *trees*: A classic algorithm by Day [1985] solves the problem in $O(n)$ time and $O(n)$ space.

New results in this paper:

- $O(n\,e/\log n)$ time and $O(n\,m/\log n)$ words, assuming the word RAM model with word length $\omega = \lceil \log n \rceil$ bits.
- $O(n\,m/\log n)$ time and $O(n\,m/\log n)$ words for networks with bounded degree.
- $O((k+1)\,e)$ time and $O((k+1)\,m\log n)$ bits for level-k networks.
- $O(m)$ time and $O(m\log n)$ bits for leaf-outerplanar networks.

Preliminaries

Let $N = (V, E)$ be a phylogenetic network.
Recall that:

- For any $v \in V$, $C(v) =$ the set of leaves which are descendants of v.
- *The cluster collection of N* is $C(N) = \left\{ C(v) : v \in V \right\}$.

Preliminaries

Let $N = (V, E)$ be a phylogenetic network.
Recall that:

- For any $v \in V$, $C(v) =$ the set of leaves which are descendants of v.
- *The cluster collection of N* is $C(N) = \{C(v) : v \in V\}$.

Observation:
Given the cluster collections $C_1 = C(N_1)$, $C_2 = C(N_2)$ of two phylogenetic networks N_1, N_2, we can compute the Robinson-Foulds distance $d_{RF}(N_1, N_2)$ with the following algorithm.

Algorithm to compute $d_{RF}(N_1, N_2)$, given C_1, C_2

```
**function** robinson_foulds_distance($N_1, N_2$)
    radix sort $C_1$
    radix sort $C_2$
```

Algorithm to compute $d_{RF}(N_1, N_2)$, given C_1, C_2

```
function robinson_foulds_distance($N_1, N_2$)
    radix sort $C_1$
    radix sort $C_2$
    $m_1, m_2 \leftarrow$ number of nodes of $N_1, N_2$
    $i_1 \leftarrow 1$
    $i_2 \leftarrow 1$
    $c \leftarrow 0$
    while $i_1 \leq m_1$ and $i_2 \leq m_2$ do
```

Algorithm to compute $d_{RF}(N_1, N_2)$, given C_1, C_2

```
function robinson_foulds_distance($N_1, N_2$)
    radix sort $C_1$
    radix sort $C_2$
    $m_1, m_2 \leftarrow$ number of nodes of $N_1, N_2$
    $i_1 \leftarrow 1$
    $i_2 \leftarrow 1$
    $c \leftarrow 0$
    while $i_1 \leq m_1$ and $i_2 \leq m_2$ do
        if $C_1[i_1] < C_2[i_2]$ then
            $i_1 \leftarrow i_1 + 1$
        else if $C_1[i_1] > C_2[i_2]$ then
            $i_2 \leftarrow i_2 + 1$
        else
            $i_1 \leftarrow i_1 + 1$
            $i_2 \leftarrow i_2 + 1$
            $c \leftarrow c + 1$
    return $(m_1 + m_2 - 2 \cdot c)/2$
```

Cluster representations

The time complexity of the algorithm depends on *how* the clusters are represented.

Cluster representations

The time complexity of the algorithm depends on *how* the clusters are represented.

We now consider three different ways to represent a cluster collection $C(N)$:

1. Naive cluster representation.
2. Cluster representation by characteristic vectors.
3. Cluster representation by interval lists.

Cluster representations

The time complexity of the algorithm depends on *how* the clusters are represented.

We now consider three different ways to represent a cluster collection $C(N)$:

1. Naive cluster representation.
2. Cluster representation by characteristic vectors.
3. Cluster representation by interval lists.

From here on:

n = Number of leaves in N
m = Total number of nodes in N
e = Number of edges in N

1. Naive cluster representation

Explicitly store the set $C(v)$ for each $v \in V$.

1. Naive cluster representation, cont.

Cardona *et al.* [2009] compute $C(N)$ by breadth-first search from each node v to find the cluster $C(v)$. $\Rightarrow O(m\,e)$ time, $O(n\,m)$ space

1. Naive cluster representation, cont.

Cardona *et al.* [2009] compute $C(N)$ by breadth-first search from each node v to find the cluster $C(v)$. $\Rightarrow O(m\, e)$ time, $O(n\, m)$ space

Slightly faster: bottom-up traversal $\Rightarrow O(n\, e)$ time, $O(nm)$ space

```
procedure naive_cluster_representation($N$, $C$)
```

1. Naive cluster representation, cont.

Cardona *et al.* [2009] compute $C(N)$ by breadth-first search from each node v to find the cluster $C(v)$. $\Rightarrow$ $O(m\,e)$ time, $O(n\,m)$ space

Slightly faster: bottom-up traversal $\Rightarrow$ $O(n\,e)$ time, $O(nm)$ space

```
procedure naive_cluster_representation($N$, $C$)
    for each node $v$ of $N$ do
        if $v$ is a leaf then
            $C(v) \leftarrow \{label(v)\}$
            $enqueue(Q, v)$
        else
            $C(v) \leftarrow \emptyset$
```

1. Naive cluster representation, cont.

Cardona *et al.* [2009] compute $C(N)$ by breadth-first search from each node v to find the cluster $C(v)$. $\Rightarrow O(m\,e)$ time, $O(n\,m)$ space

Slightly faster: bottom-up traversal $\Rightarrow O(n\,e)$ time, $O(nm)$ space

```
procedure naive_cluster_representation($N$, $C$)
    for each node $v$ of $N$ do
        if $v$ is a leaf then
            $C(v) \leftarrow \{label(v)\}$
            $enqueue(Q, v)$
        else
            $C(v) \leftarrow \emptyset$
    while $Q$ is not empty do
        $v \leftarrow dequeue(Q)$
        mark node $v$ as visited
        for each parent $u$ of node $v$ do
            $C(u) \leftarrow C(u) \cup C(v)$
            if all children of $u$ are visited then
                $enqueue(Q, u)$
```

2. Cluster representation by characteristic vectors

Let $N = (V, E)$ be a phylogenetic network.

- Leaf numbering function:
  Bijection from the set of leaves in N to the set $\{1, 2, \ldots, n\}$.

2. Cluster representation by characteristic vectors

Let $N = (V, E)$ be a phylogenetic network.

- Leaf numbering function:
  Bijection from the set of leaves in N to the set $\{1, 2, \ldots, n\}$.

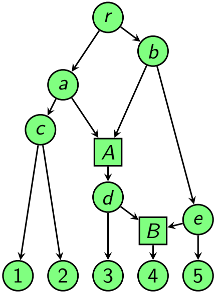
- Let f be a leaf numbering function and $v \in V$.
  The *characteristic vector for v under f* is a bit vector $C_f[v]$ of length n such that for any $i \in \{1, 2, \ldots, n\}$, the ith bit is 1 iff $f^{-1}(i)$ is a descendant of v in N.

2. Cluster representation by characteristic vectors

Let $N = (V, E)$ be a phylogenetic network.

- Leaf numbering function:
  Bijection from the set of leaves in N to the set $\{1, 2, \ldots, n\}$.

- Let f be a leaf numbering function and $v \in V$.
  The *characteristic vector for v under f* is a bit vector $C_f[v]$ of length n such that for any $i \in \{1, 2, \ldots, n\}$, the ith bit is 1 iff $f^{-1}(i)$ is a descendant of v in N.



r	11111	b	00111
a	11110	A	00110
c	11000	d	00110
e	00011	B	00010
1	10000	4	00010
2	01000	5	00001
3	00100		

2. Cluster representation by characteristic vectors

Let $N = (V, E)$ be a phylogenetic network.

- Leaf numbering function:
  Bijection from the set of leaves in N to the set $\{1, 2, \ldots, n\}$.

- Let f be a leaf numbering function and $v \in V$.
  The *characteristic vector for v under f* is a bit vector $C_f[v]$ of length n such that for any $i \in \{1, 2, \ldots, n\}$, the ith bit is 1 iff $f^{-1}(i)$ is a descendant of v in N.

- Note that $C_f[r] = 11\ldots1$ for the root r of N, and $C_f[\ell]$ contains exactly one 1 for any leaf ℓ in N.

2. Cluster representation by characteristic vectors

Let $N = (V, E)$ be a phylogenetic network.

- Leaf numbering function:
  Bijection from the set of leaves in N to the set $\{1, 2, \ldots, n\}$.

- Let f be a leaf numbering function and $v \in V$.
  The *characteristic vector for v under f* is a bit vector $C_f[v]$ of length n such that for any $i \in \{1, 2, \ldots, n\}$, the ith bit is 1 iff $f^{-1}(i)$ is a descendant of v in N.

- Note that $C_f[r] = 11\ldots1$ for the root r of N, and $C_f[\ell]$ contains exactly one 1 for any leaf ℓ in N.

Next: How can we compute $C_f[v]$ for all $v \in V$ efficiently?

2. Cluster representation by characteristic vectors, cont.

- Select any arbitrary leaf numbering function f,
  encode each cluster $C(v)$ by a characteristic vector $C_f[v]$ of n bits.

2. Cluster representation by characteristic vectors, cont.

- Select any arbitrary leaf numbering function f,
  encode each cluster $C(v)$ by a characteristic vector $C_f[v]$ of n bits.
- Then:
  The set union operation for two clusters can be implemented by taking the bitwise OR of their two bit vectors.

2. Cluster representation by characteristic vectors, cont.

- Select any arbitrary leaf numbering function f,
  encode each cluster $C(v)$ by a characteristic vector $C_f[v]$ of n bits.
- Then:
  The set union operation for two clusters can be implemented by taking the bitwise OR of their two bit vectors.
- Use the word RAM model with word length $\omega = \lceil \log n \rceil$ bits, and pack each characteristic vector into $k = \lceil n/\omega \rceil = O(n/\log n)$ words.

2. Cluster representation by characteristic vectors, cont.

- Select any arbitrary leaf numbering function f,
  encode each cluster $C(v)$ by a characteristic vector $C_f[v]$ of n bits.

- Then:
  The set union operation for two clusters can be implemented by taking the bitwise OR of their two bit vectors.

- Use the word RAM model with word length $\omega = \lceil \log n \rceil$ bits, and pack each characteristic vector into $k = \lceil n/\omega \rceil = O(n/\log n)$ words.

- Apply the same bottom-up technique as before, with preprocessing: store the bitwise OR of every pair of $\frac{\omega}{2}$-bit vectors in a table of size $2^{\omega/2} \cdot 2^{\omega/2} = O(\sqrt{n}) \cdot O(\sqrt{n}) = O(n)$ words.
  $\Rightarrow$ The union of any two clusters can be obtained in $O(n/\log n)$ time.

2. Cluster representation by characteristic vectors, cont.

- Select any arbitrary leaf numbering function f, encode each cluster $C(v)$ by a characteristic vector $C_f[v]$ of n bits.
- Then:
  The set union operation for two clusters can be implemented by taking the bitwise OR of their two bit vectors.
- Use the word RAM model with word length $\omega = \lceil \log n \rceil$ bits, and pack each characteristic vector into $k = \lceil n/\omega \rceil = O(n/\log n)$ words.
- Apply the same bottom-up technique as before, with preprocessing: store the bitwise OR of every pair of $\frac{\omega}{2}$-bit vectors in a table of size $2^{\omega/2} \cdot 2^{\omega/2} = O(\sqrt{n}) \cdot O(\sqrt{n}) = O(n)$ words.
  $\Rightarrow$ The union of any two clusters can be obtained in $O(n/\log n)$ time.

Theorem 1

In total, the cluster collection of N can be computed in $O(n\,e/\log n)$ time using $O(n\,m/\log n)$ words.

3. Cluster representation by interval lists

Interval = maximal consecutive sequence of 1's in a bit vector.

We can encode each cluster $C(v)$ of a phylogenetic network N by fixing any leaf numbering function f and storing the starting & ending positions of all intervals in the characteristic vector $C_f[v]$ in sorted order.

3. Cluster representation by interval lists

Interval = maximal consecutive sequence of 1's in a bit vector.

We can encode each cluster $C(v)$ of a phylogenetic network N by fixing any leaf numbering function f and storing the starting & ending positions of all intervals in the characteristic vector $C_f[v]$ in sorted order.

Let f be a leaf numbering function for a phylogenetic network $N = (V, E)$.

3. Cluster representation by interval lists

Interval = maximal consecutive sequence of 1's in a bit vector.

We can encode each cluster $C(v)$ of a phylogenetic network N by fixing any leaf numbering function f and storing the starting & ending positions of all intervals in the characteristic vector $C_f[v]$ in sorted order.

Let f be a leaf numbering function for a phylogenetic network $N = (V, E)$.

- For each node $v \in V$, let $I_f(v)$ be the number of intervals in $C_f[v]$.

3. Cluster representation by interval lists

Interval = maximal consecutive sequence of 1's in a bit vector.

We can encode each cluster $C(v)$ of a phylogenetic network N by fixing any leaf numbering function f and storing the starting & ending positions of all intervals in the characteristic vector $C_f[v]$ in sorted order.

Let f be a leaf numbering function for a phylogenetic network $N = (V, E)$.

- For each node $v \in V$, let $I_f(v)$ be the number of intervals in $C_f[v]$.
- The *spread of f* is $I_f = \max_{v \in V} I_f(v)$.

3. Cluster representation by interval lists

Interval = maximal consecutive sequence of 1's in a bit vector.

We can encode each cluster $C(v)$ of a phylogenetic network N by fixing any leaf numbering function f and storing the starting & ending positions of all intervals in the characteristic vector $C_f[v]$ in sorted order.

Let f be a leaf numbering function for a phylogenetic network $N = (V, E)$.

- For each node $v \in V$, let $I_f(v)$ be the number of intervals in $C_f[v]$.
- The *spread of f* is $I_f = \max_{v \in V} I_f(v)$.

Lemma 6

Given any leaf numbering function f, the total space needed to store all characteristic vectors under f using the interval list representation is $O(I_f \, m \log n)$ bits.

3. Cluster representation by interval lists, cont.

Lemma 7

Given any leaf numbering function f, the interval lists for all clusters in N can be computed in $O(I_f \cdot e)$ time.

(Again, use the bottom-up technique.
To implement the cluster union operation $C(u) \cup C(v)$, scan the two sorted interval lists for $C_f[u]$ and $C_f[v]$ and merge intervals which overlap or are immediate neighbors.)

Bounding the minimum spread

The *minimum spread of N* is the minimum value of I_f, taken over all possible leaf numbering functions f.

Bounding the minimum spread

The *minimum spread of* N is the minimum value of I_f, taken over all possible leaf numbering functions f.

According to Lemmas 6 and 7, the space needed to store the cluster collection of a phylogenetic network N using interval lists, as well as the time needed to compute these lists, depend on the minimum spread of N.

Bounding the minimum spread

The *minimum spread of* N is the minimum value of I_f, taken over all possible leaf numbering functions f.

According to Lemmas 6 and 7, the space needed to store the cluster collection of a phylogenetic network N using interval lists, as well as the time needed to compute these lists, depend on the minimum spread of N.

Next, we show that for certain classes of phylogenetic networks, we can bound the minimum spread efficiently.

Level-k phylogenetic networks

The *level* of a phylogenetic network N is a parameter that indicates how tree-like the network is.

$$\begin{cases} \text{Level-0:} & \text{Tree.} \\ \text{Level-1:} & \text{All cycles in the underlying undirected graph are disjoint.} \\ \text{Level-2:} & \text{More complicated structure.} \\ \text{etc.} \end{cases}$$

Level-k phylogenetic networks

The *level* of a phylogenetic network N is a parameter that indicates how tree-like the network is.

$$\begin{cases} \text{Level-0:} & \text{Tree.} \\ \text{Level-1:} & \text{All cycles in the underlying undirected graph are disjoint.} \\ \text{Level-2:} & \text{More complicated structure.} \\ \text{etc.} & \end{cases}$$

Introduced by Choy, Jansson, Sadakane, Sung [2005].

Level-k phylogenetic networks

The *level* of a phylogenetic network N is a parameter that indicates how tree-like the network is.

Level-0: Tree.
Level-1: All cycles in the underlying undirected graph are disjoint.
Level-2: More complicated structure.
etc.

Introduced by Choy, Jansson, Sadakane, Sung [2005].

Definition

- Let $\mathcal{U}(N)$ denote the undirected graph obtained by replacing every directed edge in N by an undirected edge.

Level-k phylogenetic networks

The *level* of a phylogenetic network N is a parameter that indicates how tree-like the network is.

$$\begin{cases} \text{Level-0:} & \text{Tree.} \\ \text{Level-1:} & \text{All cycles in the underlying undirected graph are disjoint.} \\ \text{Level-2:} & \text{More complicated structure.} \\ \text{etc.} & \end{cases}$$

Introduced by Choy, Jansson, Sadakane, Sung [2005].

Definition

- Let $\mathcal{U}(N)$ denote the undirected graph obtained by replacing every directed edge in N by an undirected edge.
- A *biconnected component* of an undirected graph is a connected subgraph that remains connected after deleting any node.

Level-k phylogenetic networks

The *level* of a phylogenetic network N is a parameter that indicates how tree-like the network is.

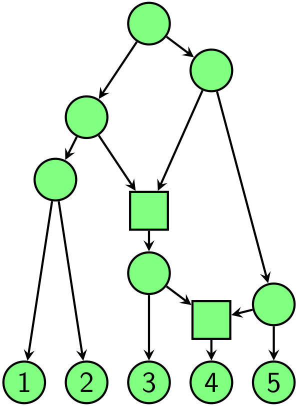
Level-0: Tree.
Level-1: All cycles in the underlying undirected graph are disjoint.
Level-2: More complicated structure.
etc.

Introduced by Choy, Jansson, Sadakane, Sung [2005].

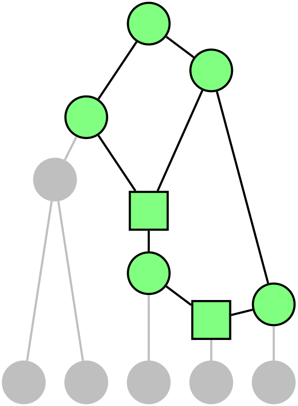
Definition

- Let $\mathcal{U}(N)$ denote the undirected graph obtained by replacing every directed edge in N by an undirected edge.
- A *biconnected component* of an undirected graph is a connected subgraph that remains connected after deleting any node.
- N is a level-k phylogenetic network if, for every biconnected component B in $\mathcal{U}(N)$, the subgraph of N induced by the set of nodes in B contains at most k hybrid nodes.

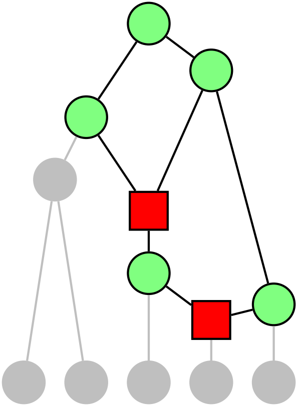
Level-*k* phylogenetic network, example



Level-*k* phylogenetic network, example

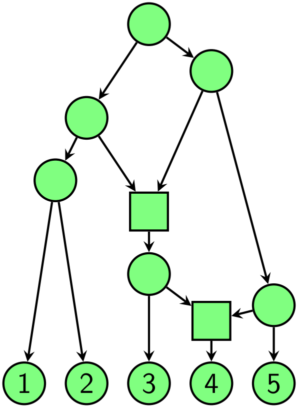


Level-k phylogenetic network, example



Every biconnected component in $\mathcal{U}(N)$ has at most 2 nodes that are hybrid nodes in N.

Level-k phylogenetic network, example



Every biconnected component in $\mathcal{U}(N)$ has at most 2 nodes that are hybrid nodes in N.

$\Rightarrow$ N is a level-2 network.

The minimum spread of a level-k phylogenetic network

Lemma 5

If N is a level-k phylogenetic network then a leaf numbering function f with $l_f \leq k+1$ exists and can be computed in $O(e)$ time.

The minimum spread of a level-k phylogenetic network

Lemma 5

If N is a level-k phylogenetic network then a leaf numbering function f with $l_f \leq k+1$ exists and can be computed in $O(e)$ time.

Proof:

Fix any directed spanning tree T of N.
Let f be the leaf numbering obtained by a depth-first search of T starting at the root, assigning $1, 2, \ldots, n$ to the leaves in the order they are visited.

The minimum spread of a level-k phylogenetic network

Lemma 5

If N is a level-k phylogenetic network then a leaf numbering function f with $l_f \leq k+1$ exists and can be computed in $O(e)$ time.

Proof:

Fix any directed spanning tree T of N.
Let f be the leaf numbering obtained by a depth-first search of T starting at the root, assigning $1, 2, \ldots, n$ to the leaves in the order they are visited.

For every node v in N, define $L(T[v]) =$ the set of leaves in the subtree of T rooted at v.

The minimum spread of a level-k phylogenetic network

Lemma 5

If N is a level-k phylogenetic network then a leaf numbering function f with $I_f \leq k + 1$ exists and can be computed in $O(e)$ time.

Proof:

Fix any directed spanning tree T of N.
Let f be the leaf numbering obtained by a depth-first search of T starting at the root, assigning $1, 2, \ldots, n$ to the leaves in the order they are visited.

For every node v in N, define $L(T[v])$ = the set of leaves in the subtree of T rooted at v.

Key observation:
For every node v, the leaves belonging to $L(T[v])$ are visited consecutively by any depth-first search of T.
$\Rightarrow$ These leaves form a single interval in $C_f[v]$.

(cont.$\longrightarrow$)

Proof: (→ cont.)

Next, consider any node u in N.

Let $H =$ the set of all hybrid nodes in N that:

- Belong to the same biconnected component as u, and
- Are proper descendants of u.

(The set H may be empty.)

Proof: (→ cont.)

Next, consider any node u in N.

Let $H =$ the set of all hybrid nodes in N that:

- Belong to the same biconnected component as u, and
- Are proper descendants of u.

(The set H may be empty.)

Then, the set of leaves that are descendants of u in N can be written as: $\boxed{L(T[u]) \cup \bigcup_{h \in H} L(T[h])}$.

Proof: ($\rightarrow$ cont.)
Next, consider any node u in N.
Let H = the set of all hybrid nodes in N that:

- Belong to the same biconnected component as u, and
- Are proper descendants of u.

(The set H may be empty.)

Then, the set of leaves that are descendants of u in N can be written as: $\boxed{L(T[u]) \cup \bigcup_{h \in H} L(T[h])}$.

N is a level-k phylogenetic network, so $|H| \leq k$.

Proof: (→ cont.)

Next, consider any node u in N.

Let $H =$ the set of all hybrid nodes in N that:

- Belong to the same biconnected component as u, and
- Are proper descendants of u.

(The set H may be empty.)

Then, the set of leaves that are descendants of u in N can be written as: $\boxed{L(T[u]) \cup \bigcup_{h \in H} L(T[h])}$.

N is a level-k phylogenetic network, so $|H| \leq k$.

By the key observation, each subset $L(T[v])$ of leaves forms one interval in $C_f[v]$.

Thus, $C_f[u]$ is the union of at most $k+1$ intervals.

Proof: (→ cont.)

Next, consider any node u in N.

Let $H =$ the set of all hybrid nodes in N that:

- Belong to the same biconnected component as u, and
- Are proper descendants of u.

(The set H may be empty.)

Then, the set of leaves that are descendants of u in N can be written as: $\boxed{L(T[u]) \cup \bigcup_{h \in H} L(T[h])}$.

N is a level-k phylogenetic network, so $|H| \leq k$.

By the key observation, each subset $L(T[v])$ of leaves forms one interval in $C_f[v]$.

Thus, $C_f[u]$ is the union of at most $k+1$ intervals.

It follows that $I_f(u) \leq k+1$ for every $u \in V$,
i.e., the spread of f is $I_f = \max_{u \in V} I_f(u) \leq k+1$. □

Computing $C(N)$ for a level-k network

We have just proved:

Lemma 5

If N is a level-k phylogenetic network then a leaf numbering function f with $l_f \leq k+1$ exists and can be computed in $O(e)$ time.

Computing $C(N)$ for a level-k network

We have just proved:

Lemma 5

If N is a level-k phylogenetic network then a leaf numbering function f with $l_f \leq k+1$ exists and can be computed in $O(e)$ time.

Now, applying Lemmas 6 and 7 immediately gives:

Theorem 2

If N is a level-k phylogenetic network, the cluster collection $C(N)$ of N can be computed in $O((k+1)\, e)$ time and $O((k+1)\, m \log n)$ bits.

Leaf-outerplanar phylogenetic networks

By Lemma 5, every level-k network has minimum spread $\leq k+1$.
What about other classes of structurally restricted networks?

Leaf-outerplanar phylogenetic networks

By Lemma 5, every level-k network has minimum spread $\leq k + 1$.
What about other classes of structurally restricted networks?

Definition

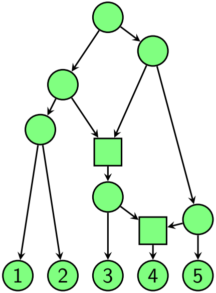
A phylogenetic network N is leaf-outerplanar if $\mathcal{U}(N)$ admits a non-crossing layout in the plane with the root and all leaves on the outer face.

Leaf-outerplanar phylogenetic networks

By Lemma 5, every level-k network has minimum spread $\leq k + 1$.
What about other classes of structurally restricted networks?

Definition

A phylogenetic network N is leaf-outerplanar if $\mathcal{U}(N)$ admits a non-crossing layout in the plane with the root and all leaves on the outer face.



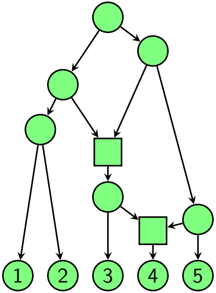
Leaf-outerplanar

Leaf-outerplanar phylogenetic networks

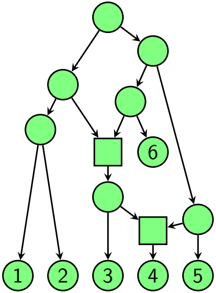
By Lemma 5, every level-k network has minimum spread $\leq k+1$.
What about other classes of structurally restricted networks?

Definition

A phylogenetic network N is leaf-outerplanar if $\mathcal{U}(N)$ admits a non-crossing layout in the plane with the root and all leaves on the outer face.



Leaf-outerplanar



Not leaf-outerplanar

Leaf-outerplanar phylogenetic networks

By Lemma 5, every level-k network has minimum spread $\leq k+1$.
What about other classes of structurally restricted networks?

Definition

A phylogenetic network N is leaf-outerplanar if $\mathcal{U}(N)$ admits a non-crossing layout in the plane with the root and all leaves on the outer face.

Useful concept because:
Leaf-outerplanar phylogenetic networks are output by certain phylogenetic network construction methods such as Neighbor-Net (Bryant & Moulton [2004]) and QNet (Grünewald *et al.* [2007]).

The minimum spread of a leaf-outerplanar network

Lemma 4

If N is a leaf-outerplanar phylogenetic network then a leaf numbering function f with $I_f = 1$ exists and can be computed in $O(m)$ time.

The minimum spread of a leaf-outerplanar network

Lemma 4

If N is a leaf-outerplanar phylogenetic network then a leaf numbering function f with $l_f = 1$ exists and can be computed in $O(m)$ time.

Proof: Join the root & all leaves in N to a new vertex, and run the linear-time planar embedding algorithm of Chiba *et al.* [1985] to construct some leaf-outerplanar embedding for N.

The minimum spread of a leaf-outerplanar network

Lemma 4

If N is a leaf-outerplanar phylogenetic network then a leaf numbering function f with $I_f = 1$ exists and can be computed in $O(m)$ time.

Proof: Join the root & all leaves in N to a new vertex, and run the linear-time planar embedding algorithm of Chiba *et al.* [1985] to construct some leaf-outerplanar embedding for N.
Let f assign $\{1, 2, \ldots, n\}$ to the leaves consecutively along the outer face.

The minimum spread of a leaf-outerplanar network

Lemma 4

If N is a leaf-outerplanar phylogenetic network then a leaf numbering function f with $I_f = 1$ exists and can be computed in $O(m)$ time.

Proof: Join the root & all leaves in N to a new vertex, and run the linear-time planar embedding algorithm of Chiba *et al.* [1985] to construct some leaf-outerplanar embedding for N.
Let f assign $\{1, 2, \ldots, n\}$ to the leaves consecutively along the outer face.

Consider any node v in N. Two cases:

- v is a leaf: Trivially, $C_f[v]$ has a single interval.

The minimum spread of a leaf-outerplanar network

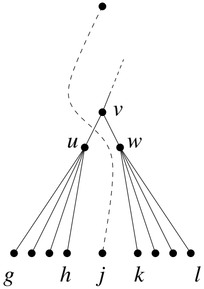
Lemma 4

If N is a leaf-outerplanar phylogenetic network then a leaf numbering function f with $I_f = 1$ exists and can be computed in $O(m)$ time.

Proof: Join the root & all leaves in N to a new vertex, and run the linear-time planar embedding algorithm of Chiba *et al.* [1985] to construct some leaf-outerplanar embedding for N.
Let f assign $\{1, 2, \ldots, n\}$ to the leaves consecutively along the outer face.

Consider any node v in N. Two cases:

- v is a leaf: Trivially, $C_f[v]$ has a single interval.
- v is an internal node: Suppose u, w are children of v and $C(u) = \{g, \ldots, h\}$, $C(w) = \{k, \ldots, \ell\}$, $i, \ldots, j \notin C(v)$, but $f(g) \leq f(h) < f(i) \leq f(j) < f(k) \leq f(\ell)$. A path from the root to a leaf in $\{i, \ldots, j\}$ may not pass through v; hence it crosses either the path from v to h or the path from v to k. Contradiction.



The minimum spread of a leaf-outerplanar network

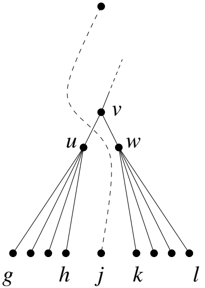
Lemma 4

If N is a leaf-outerplanar phylogenetic network then a leaf numbering function f with $I_f = 1$ exists and can be computed in $O(m)$ time.

Proof: Join the root & all leaves in N to a new vertex, and run the linear-time planar embedding algorithm of Chiba *et al.* [1985] to construct some leaf-outerplanar embedding for N.
Let f assign $\{1, 2, \ldots, n\}$ to the leaves consecutively along the outer face.

Consider any node v in N. Two cases:

- v is a leaf: Trivially, $C_f[v]$ has a single interval.
- v is an internal node: Suppose u, w are children of v and $C(u) = \{g, \ldots, h\}$, $C(w) = \{k, \ldots, \ell\}$, $i, \ldots, j \notin C(v)$, but $f(g) \leq f(h) < f(i) \leq f(j) < f(k) \leq f(\ell)$. A path from the root to a leaf in $\{i, \ldots, j\}$ may not pass through v; hence it crosses either the path from v to h or the path from v to k. Contradiction.



⇒ For every $v \in V$, $C_f[v]$ has a single interval. □

Computing $C(N)$ for a leaf-outerplanar network

Lemma 4

If N is a leaf-outerplanar phylogenetic network then a leaf numbering function f with $I_f = 1$ exists and can be computed in $O(m)$ time.

Computing $C(N)$ for a leaf-outerplanar network

Lemma 4

If N is a leaf-outerplanar phylogenetic network then a leaf numbering function f with $I_f = 1$ exists and can be computed in $O(m)$ time.

This time, applying Lemmas 6 and 7 gives:

Theorem 3

If N is a leaf-outerplanar phylogenetic network, the cluster collection $C(N)$ of N can be computed in $O(m)$ time and $O(m \log n)$ bits.

Summary

New results in this paper:

We can compute the Robinson-Foulds distance $d_{RF}(N_1, N_2)$ of two phylogenetic networks N_1, N_2 in:

- $O(n\,e/\log n)$ time and $O(n\,m/\log n)$ words, assuming the word RAM model with word length $\omega = \lceil \log n \rceil$ bits.
- $O(n\,m/\log n)$ time and $O(n\,m/\log n)$ words for networks with bounded degree.
- $O((k+1)\,e)$ time and $O((k+1)\,m \log n)$ bits for level-k networks.
- $O(m)$ time and $O(m \log n)$ bits for leaf-outerplanar networks.

Summary

New results in this paper:

We can compute the Robinson-Foulds distance $d_{RF}(N_1, N_2)$ of two phylogenetic networks N_1, N_2 in:

- $O(n\,e/\log n)$ time and $O(n\,m/\log n)$ words, assuming the word RAM model with word length $\omega = \lceil \log n \rceil$ bits.
- $O(n\,m/\log n)$ time and $O(n\,m/\log n)$ words for networks with bounded degree.
- $O((k+1)\,e)$ time and $O((k+1)\,m \log n)$ bits for level-k networks.
- $O(m)$ time and $O(m \log n)$ bits for leaf-outerplanar networks.

- We have also introduced a new parameter called the *minimum spread* of a phylogenetic network, and shown that d_{RF} can be computed efficiently when the minimum spread is small.

Summary

New results in this paper:

We can compute the Robinson-Foulds distance $d_{RF}(N_1, N_2)$ of two phylogenetic networks N_1, N_2 in:

- $O(n\,e/\log n)$ time and $O(n\,m/\log n)$ words, assuming the word RAM model with word length $\omega = \lceil \log n \rceil$ bits.
- $O(n\,m/\log n)$ time and $O(n\,m/\log n)$ words for networks with bounded degree.
- $O((k+1)\,e)$ time and $O((k+1)\,m \log n)$ bits for level-k networks.
- $O(m)$ time and $O(m \log n)$ bits for leaf-outerplanar networks.

- We have also introduced a new parameter called the *minimum spread* of a phylogenetic network, and shown that d_{RF} can be computed efficiently when the minimum spread is small.
- In particular, the minimum spread of a level-k network is $\leq k+1$, and the minimum spread of a leaf-outerplanar network is 1.