# Faster Computation of the Robinson-Foulds Distance between Phylogenetic Networks

Tetsuo Asano[1], Jesper Jansson[2], Kunihiko Sadakane[3], Ryuhei Uehara[1], and Gabriel Valiente[4]

[1] School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan, t-asano@jaist.ac.jp, uehara@jaist.ac.jp
[2] Ochanomizu University, 2-1-1 Otsuka, Bunkyo-ku, Tokyo 112-8610, Japan, jesper.jansson@ocha.ac.jp
[3] National Institute of Informatics, Hitotsubashi 2-1-2, Chiyoda-ku, Tokyo 101-8430, Japan, sada@nii.ac.jp
[4] Algorithms, Bioinformatics, Complexity and Formal Methods Research Group, Technical University of Catalonia, E-08034 Barcelona, Spain, valiente@lsi.upc.edu

**Abstract.** The Robinson-Foulds distance, which is the most widely used metric for comparing phylogenetic trees, has recently been generalized to phylogenetic networks. Given two networks $N_1, N_2$ with $n$ leaves, $m$ nodes, and $e$ edges, the Robinson-Foulds distance measures the number of clusters of descendant leaves that are not shared by $N_1$ and $N_2$. The fastest known algorithm for computing the Robinson-Foulds distance between those networks runs in $O(m(m+e))$ time. In this paper, we improve the time complexity to $O(n(m+e)/\log n)$ for general networks and $O(n\,m/\log n)$ for general networks with bounded degree, and to optimal $O(m+e)$ time for planar phylogenetic networks and bounded-level phylogenetic networks. We also introduce the natural concept of the minimum spread of a phylogenetic network and show how the running time of our new algorithm depends on this parameter. As an example, we prove that the minimum spread of a level-$k$ phylogenetic network is at most $k+1$, which implies that for two level-$k$ phylogenetic networks, our algorithm runs in $O((k+1)(m+e))$ time.

## 1 Introduction

The Robinson-Foulds distance, introduced in [17], has been the most widely used metric over almost three decades for comparing phylogenetic trees. However, it is now known that the evolutionary history of life cannot be properly represented as a phylogenetic tree [7], and phylogenetic networks have emerged as the representation of choice for incorporating reticulate evolutionary events, like recombination, hybridization, or lateral gene transfer, in an evolutionary history [16].

Phylogenetic networks are directed acyclic graphs with *tree nodes* (those with at most one parent) corresponding to point mutation events

and *hybrid nodes* (with more than one parent) corresponding to hybrid speciation events. As in the case of phylogenetic trees, the leaves are distinctly labeled by a set of extant species. Additional conditions are usually imposed on these directed acyclic graphs to narrow down the output space of reconstruction algorithms [13, 14] or to provide a realistic model of recombination [19, 20].

Two such additional conditions are especially relevant to the Robinson-Foulds distance. A phylogenetic network is *time consistent* when it has a temporal representation [1], that is, an assignment of discrete time stamps to the nodes that increases from parents to tree children and remains the same from parents to hybrid children, meaning that the parents of each hybrid node coexist in time and thus, the corresponding reticulate evolutionary event can take place. A phylogenetic network is *tree-child* when every internal node has at least one tree child [4], meaning that every non-extant species has some extant descendant through mutation alone.

The Robinson-Foulds distance between two phylogenetic networks is defined as the cardinality of the symmetric difference between their two sets of induced clusters of descendant leaves (where the cluster induced by a node $v$ in a phylogenetic network is the set of all descendant leaves of $v$ in the network) divided by two, and thus it measures the number of clusters not shared by the networks. It is a metric on the space of all tree-child time-consistent phylogenetic networks [4, Cor. 1], and it generalizes the Robinson-Foulds distance between rooted phylogenetic trees. Clearly, the Robinson-Foulds distance requires computing the cluster representation of the networks, that is, the set of descendant leaves for each node in the networks. While there are improved algorithms for computing the cluster representation of a phylogenetic tree [6, 15, 21, 22], the only known algorithm for computing the cluster representation of a phylogenetic network [3] is based on breadth-first searching descendant leaves from each of the nodes in turn, and takes $O(m(m+e))$ time using $O(n\,m)$ space on phylogenetic networks with $n$ leaves, $m$ nodes, and $e$ edges.

In this paper, we present a faster algorithm for computing the Robinson-Foulds distance between two input phylogenetic networks. For general phylogenetic networks, we first improve the time complexity by following an approach similar in spirit to the algorithm proposed in [4] for computing the path multiplicity representation of a phylogenetic network; by using a compressed representation of the characteristic vectors, we obtain a simple algorithm for computing the Robinson-Foulds distance between phylogenetic networks in $O(n(m+e)/\log n)$ time using $O(n\,m/\log n)$

space, assuming a word size of $\omega = \lceil \log n \rceil$ bits; see [12]. For phylogenetic networks of bounded degree, this becomes $O(n\,m/\log n)$ time and space.

In the case of level-$k$ phylogenetic networks [5], we further improve the time complexity by using a more succinct representation of a cluster of descendant leaves as an interval of consecutive integers, which allows us to compute the Robinson-Foulds distance in $O((k+1)(m+e))$ time. For this purpose, we introduce what we call the *minimum spread* of a phylogenetic network, and prove that every level-$k$ network has minimum spread at most $k+1$. For special cases of bounded-level phylogenetic networks such as planar phylogenetic networks, in particular outer-labeled planar split networks [2, 8] and galled-trees [10, 11], we show that the minimum spread is 1, which means that our algorithm can be implemented to run in optimal $O(m+e)$ time.

The paper is organized as follows. Section 2 introduces some notation and explains the naive representation of clusters. Section 3 describes more efficient ways to represent the clusters both for general networks and for planar and level-$k$ networks, and defines the *minimum spread* of a phylogenetic network. A bottom-up algorithm for computing the Robinson-Foulds distance is presented in Section 4 that takes advantage of the cluster representation. Finally, some conclusions are drawn in Section 5.

## 2 Preliminaries

Let $N = (V, E)$ be a given phylogenetic network with $n$ leaves, $m$ nodes, and $e$ edges. For any nodes $u, v \in V$, we say that $v$ is a *descendant* of $u$ if $v$ is reachable from $u$ in $N$. (Here, any node is considered to be a descendant of itself.) For every $v \in V$, define $C[v]$ as the set of all leaves which are descendants of $v$. The set $C[v]$ is called the *cluster* of $v$, and the collection $\{C[v] \mid v \in V\}$ is called the *naive cluster representation* of $N$.

The naive cluster representation of $N$ can be computed in $O(m(m+e))$ time and $O(n\,m)$ space by breadth-first searching descendant leaves from each of the nodes of $N$ in turn [3]. A significant improvement in time complexity can be achieved by replacing the $m$ top-down searches by $n$ bottom-up searches, because $m$ can be arbitrarily large for a phylogenetic network with $n$ leaves and, even in the particular case of a tree-child time-consistent phylogenetic network, $m \leqslant (n+4)(n-1)/2$, and this bound is tight [3, Prop. 1]. The following lemma is the basis of such an improvement.

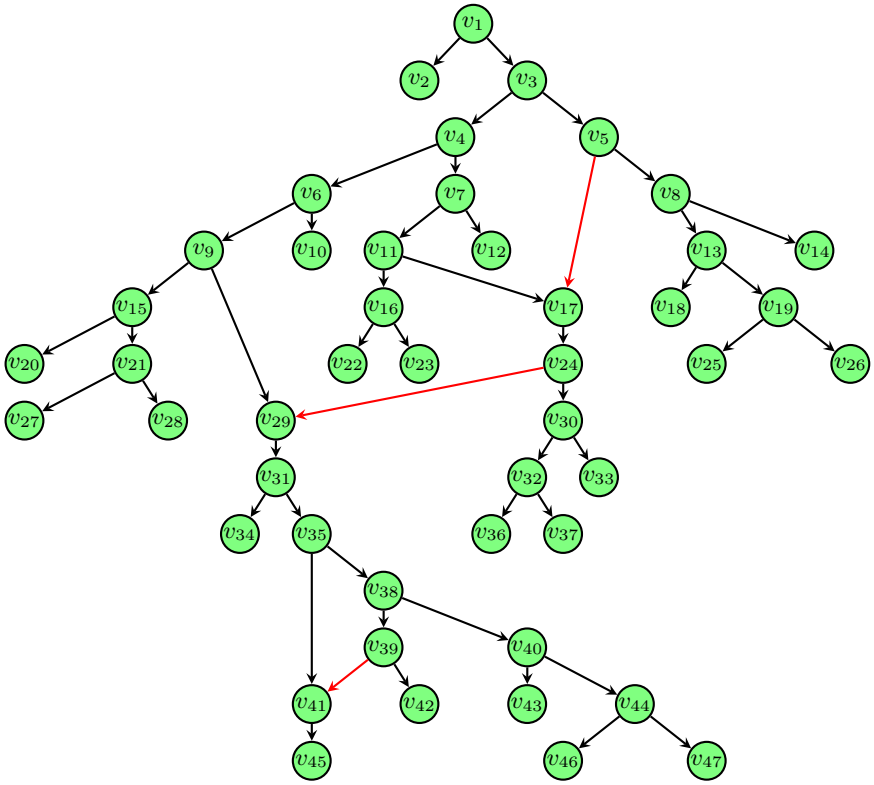


**Fig. 1.** An example of a phylogenetic network based on real data, adapted from [23]. This is the smallest level-2 phylogenetic network consistent with 1,330 rooted triplets of sequences from different isolates of the yeast *Cryptococcus gattii*.

**Lemma 1.** *Let $v \in V$ be a node of a phylogenetic network $N = (V, E)$. Then, $C[v] = \{v\}$ if $v$ is a leaf, and $C[v] = C[v_1] \cup \cdots \cup C[v_k]$ if $v$ is an internal node with children $\{v_1, \ldots, v_k\}$.*

*Proof.* The only (trivial) descendant of a leaf in a phylogenetic network is the leaf itself. The paths from an internal node to the leaves of a phylogenetic network are the paths from the children of the internal node to the leaves. □

Lemma 1 suggests a simple bottom-up algorithm (Algorithm 1) for computing the naive cluster representation of $N$ in polynomial time. In the following description, the cluster $C[v]$ of each node $v$ in $N$ is computed during a bottom-up traversal of $N$, with the help of an (initially empty)

queue $Q$ of nodes. The cluster $C[v]$ of each child $v$ of an internal node $u$ is joined in turn to the (initially empty) cluster $C[u]$ of the parent node $u$.

**Algorithm 1** *Compute the naive cluster representation $C$ of a phylogenetic network $N$.*

```
procedure naive_cluster_representation(N, C)
    for each node v of N do
        if v is a leaf then
            C[v] ← {label(v)}
            enqueue(Q, v)
        else
            C[v] ← ∅
    while Q is not empty do
        v ← dequeue(Q)
        mark node v as visited
        for each parent u of node v do
            C[u] ← C[u] ∪ C[v]
            if all children of u are visited then
                enqueue(Q, u)
```

**Lemma 2.** *Let $N$ be a phylogenetic network with $n$ leaves, $m$ nodes, and $e$ edges. The naive cluster representation of $N$ can be computed in $O(n(m+e))$ time using $O(n\,m)$ space.*

*Proof.* Each node is enqueued and dequeued only once, and each parent of each dequeued node $v$ is visited only once from $v$. The union of two subsets of an $n$ element set, which takes $O(n)$ time, is computed $O(m+e)$ times. □

## 3 More Efficient Cluster Representation

### 3.1 Characteristic Vector Representation

A *leaf numbering function* is a bijection from the set of leaves in $N$ to the set $\{1, 2, \ldots, n\}$. For any leaf numbering function $f$ and node $v \in V$, the *characteristic vector* for $v$ under $f$, denoted by $C_f[v]$, is a bit vector of length $n$ such that for any $i \in \{1, 2, \ldots, n\}$, the $i$th bit equals 1 if and only if $f^{-1}(i)$ is a descendant of $v$ in $N$. Note that $C_f[r] = 111\ldots1$ for the root $r$ of $N$, and that $C_f[\ell]$ contains exactly one 1 for any leaf $\ell$ of $N$.

**Table 1.** Characteristic vector representation of the clusters for the phylogenetic network in Figure 1.

| node | characteristic vector of the cluster | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $v_2$ | $v_{20}$ | $v_{27}$ | $v_{28}$ | $v_{34}$ | $v_{45}$ | $v_{42}$ | $v_{43}$ | $v_{46}$ | $v_{47}$ | $v_{10}$ | $v_{22}$ | $v_{23}$ | $v_{36}$ | $v_{37}$ | $v_{33}$ | $v_{12}$ | $v_{18}$ | $v_{25}$ | $v_{26}$ | $v_{14}$ |
| $v_{21}$ | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_{15}$ | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_{41}$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_{39}$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_{44}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_{40}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_{38}$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_{35}$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_{31}$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_{29}$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_9$ | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_6$ | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_{16}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_{32}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_{30}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| $v_{24}$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| $v_{17}$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| $v_{11}$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| $v_7$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| $v_4$ | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| $v_{19}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| $v_{13}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| $v_8$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| $v_5$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| $v_3$ | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| $v_1$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

*Example 1.* Consider the phylogenetic network in Figure 1. Number the leaves according to the circular ordering $v_2, v_{20}, v_{27}, v_{28}, v_{34}, v_{45}, v_{42}, v_{43}, v_{46}, v_{47}, v_{10}, v_{22}, v_{23}, v_{36}, v_{37}, v_{33}, v_{12}, v_{18}, v_{25}, v_{26}, v_{14}$ along the outer face. This corresponds to a depth-first search of the directed spanning tree obtained by removing one incoming edge (shown in red in Figure 1) for each node of in-degree 2 in the network, and it yields the characteristic vectors listed in Table 1. □

Obviously, the characteristic vector representation of all clusters in $N$ can be stored explicitly using a total of $m\,n$ bits and can be constructed in $O(n(m+e))$ time by an algorithm analogous to Algorithm 1. Our next goal is to find suitable leaf numbering functions for different types of phylogenetic networks which lead to more compact ways of storing the characteristic vectors as well as faster ways of computing them. We first consider arbitrary leaf numbering functions, and then study leaf numbering functions for some important special classes of phylogenetic networks.

### 3.2 Compressed Characteristic Vector Representation

Fix any arbitrary leaf numbering function $f$ for the given phylogenetic network $N$. The time complexity of Algorithm 1 can be improved by em-

ploying a characteristic vector of size $n$ to encode each cluster, packing the characteristic vector of a subset of the $n$ leaves into $O(n/\log n)$ integers (assuming a word size of $\omega = \lceil \log n \rceil$ bits), and computing the bitwise-OR of vectors instead of performing the set union operation. See [12] for further details about bit-level parallelism.

The pseudocode for the improved version of Algorithm 1 is given in Algorithm 2, where $x \ll t$ denotes the bitwise shift of an integer $x$ to the left by $t$, and $x \mid y$ denotes the bitwise OR of two integers $x$ and $y$.

**Algorithm 2** *Compute the compressed cluster representation $C$ of a phylogenetic network $N$.*

```
procedure compressed_cluster_representation(N, C)
    n ← number of leaves of N
    k ← ⌈n/ω⌉
    for x ← 0, ..., n − 1 do
        for y ← 0, ..., n − 1 do
            OR[x, y] ← x | y
    for each node v of N do
        C_1[v], ..., C_k[v] ← 0
        if v is a leaf then
            i ← ⌊(f(v) − 1)/ω⌋ + 1
            C_i[v] ← 1 ≪ ω · i − f(v)
            enqueue(Q, v)
    while Q is not empty do
        v ← dequeue(Q)
        mark node v as visited
        for each parent u of node v do
            for i ← 1, ..., k do
                C_i[u] ← OR[C_i[u], C_i[v]]
            if all children of u are visited then
                enqueue(Q, u)
```

The improvement in time complexity of Algorithm 2 comes from bit-level parallelism of the set union operations.

**Lemma 3.** *Let $N$ be a phylogenetic network with $n$ leaves, $m$ nodes, and $e$ edges. The cluster representation of $N$ can be computed in $O(n(m + e)/\log n)$ time using $O(n^2 + n\,m/\log n)$ words.*

*Proof.* There are $2^{\log n} = n$ bit vectors, and the bitwise-OR of all these $\omega$-bit vectors takes $O(n^2)$ time. After this prepreprocessing, each node is enqueued and dequeued only once, and each parent of each dequeued node $v$ is visited only once from $v$. The union of two subsets of an $n$ element set, which takes $O(n/\log n)$ time as the bitwise-OR of $\lceil n/\omega \rceil$ $\omega$-bit vectors, is computed $O(m+e)$ times.

The bitwise-OR of all the $\omega$-bit vectors is stored in $O(n^2)$ words, and the cluster representation is stored as a compact boolean table, with $m$ rows and $n/\log n$ columns. □

### 3.3 Interval List Representation

A maximal consecutive sequence of 1's in a bit vector is called an *interval*. For a given leaf numbering function $f$ and node $v \in V$, let $I_f(v)$ denote the number of intervals in $C_f[v]$ and let the *spread of* $f$ be $I_f = \max_{v \in V} I_f(v)$. The *minimum spread of* $N$ is the minimum value of $I_f$, taken over all possible leaf numbering functions $f$.

Below, we first bound the minimum spread of certain types of phylogenetic networks, and then show more generally how the characteristic vectors of phylogenetic networks having small minimum spread can be stored compactly. From here on, we only consider phylogenetic networks in which each node has either at most one parent (tree node) or exactly two parents (hybrid node).

A phylogenetic network is *planar* if the underlying undirected graph is outer-labeled planar, that is, if it admits a non-crossing layout on the plane with all the leaves lying on the outer face. Planar phylogenetic networks arise for instance when representing conflicting phylogenetic signals, leading to the so-called outer-labeled planar split networks; see [2, 9].

**Lemma 4.** *If $N$ is a planar phylogenetic network then a leaf numbering function $f$ with $I_f = 1$ can be computed in $O(m+e)$ time.*

*Proof.* Fix any planar embedding of $N$ and let $f$ be the leaf numbering function that assigns the numbers $1, 2, \ldots, n$ to the leaves in consecutive order along the outer face from the leftmost to the rightmost leaf. We claim that for every $v \in V$, $C_f[v]$ has a single interval. Since every leaf has a singleton cluster and the union of two overlapping or neighboring intervals is a single interval, we need to show that the children of any internal node have overlapping or neighboring clusters of descendant leaves.

Let $v \in V$ be an internal node with children $u, w \in V$ and assume $C[u] = \{h, \ldots, i\}$ and $C[w] = \{\ell, \ldots, m\}$ are intervals of descendant leaves with $h \leqslant i < j \leqslant k < \ell \leqslant m$ but $j, \ldots, k \notin C[v]$. Then, any path from the root of $N$ to any of the leaves $j, \ldots, k$ will cross some edge along either a path from $v$ to $i$ or a path from $v$ to $\ell$, contradicting the assumption that $N$ is planar. Therefore, $j, \ldots, k \in C[v]$ and the set $\{h, \ldots, i, j, \ldots, k, \ell, \ldots, m\}$ of descendant leaves forms one interval. □

Next, 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 removing any node and all edges incident to it; see [18]. Recall the following definition from [5].

**Definition 1.** *A network $N$ is called* 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 $k$ nodes with indegree 2.*

**Corollary 1.** *If $N$ is a level-1 phylogenetic network (that is, a galled-tree [10, 11]), then a leaf numbering function $f$ with $I_f = 1$ can be computed in $O(m + e)$ time.*

*Proof.* Since each biconnected component of $N$ forms a cycle and all the cycles in $N$ are disjoint, the outside of an embedding of a cycle into a plane lies on the outer-plane. Then, it is obvious that $I_f = 1$. □

**Lemma 5.** *If $N$ is a level-$k$ phylogenetic network then a leaf numbering function $f$ with $I_f = k + 1$ can be computed in $O(m + e)$ time.*

*Proof.* Fix any (directed) spanning tree $T$ of $N$, and let $f$ be the leaf numbering function obtained by doing a depth-first search of $T$ starting at the root and assigning the numbers $1, 2, \ldots, n$ to the leaves in the order that they are first visited. Clearly, this takes $O(m + e)$ time.

We now prove that $f$ has spread $k + 1$. For any node $v$ in $V$, define $L(T[v])$ as the set of all leaves in the subtree of $T$ rooted at $v$. The key observation is that the leaves in $L(T[v])$ must be visited consecutively by any depth-first search of $T$, and thus form a single interval in $C_f[v]$. Next, let $u$ be any node in $V$ and let $H$ be the set of hybrid nodes in $N$ that belong to the same biconnected component as $u$ and which are descendants of $u$ (in case $u$ is not on any merge path then $H$ is the empty set). Then, the set of leaves that are descendants of $u$ in $N$ can be written as $L(T[u]) \cup \bigcup_{h \in H} L(T[h])$. $N$ is a level-$k$ phylogenetic network,

so $|H| \leqslant k$, which together with the key observation above implies that $C_f[u]$ is the union of at most $k+1$ intervals. It follows that $I_f(u) \leqslant k+1$ for every $u \in V$. □

**Table 2.** Interval list representation of the clusters for the phylogenetic network in Figure 1.

| node | interval list |
|---|---|
| $v_{21}$ | $(v_{27}, v_{28})$ |
| $v_{15}$ | $(v_{20}, v_{28})$ |
| $v_{41}$ | $(v_{45}, v_{45})$ |
| $v_{39}$ | $(v_{45}, v_{42})$ |
| $v_{44}$ | $(v_{46}, v_{47})$ |
| $v_{40}$ | $(v_{43}, v_{47})$ |
| $v_{38}$ | $(v_{45}, v_{47})$ |
| $v_{35}$ | $(v_{45}, v_{47})$ |
| $v_{31}$ | $(v_{34}, v_{47})$ |
| $v_{29}$ | $(v_{34}, v_{47})$ |

| node | interval list |
|---|---|
| $v_9$ | $(v_{20}, v_{47})$ |
| $v_6$ | $(v_{20}, v_{10})$ |
| $v_{16}$ | $(v_{22}, v_{23})$ |
| $v_{32}$ | $(v_{36}, v_{37})$ |
| $v_{16}$ | $(v_{22}, v_{23})$ |
| $v_{32}$ | $(v_{36}, v_{37})$ |
| $v_{30}$ | $(v_{36}, v_{33})$ |
| $v_{24}$ | $(v_{34}, v_{47}), (v_{36}, v_{33})$ |
| $v_{17}$ | $(v_{34}, v_{47}), (v_{36}, v_{33})$ |
| $v_{11}$ | $(v_{34}, v_{47}), (v_{22}, v_{33})$ |

| node | interval list of the cluster |
|---|---|
| $v_7$ | $(v_{34}, v_{47}), (v_{22}, v_{12})$ |
| $v_4$ | $(v_{20}, v_{12})$ |
| $v_{19}$ | $(v_{25}, v_{26})$ |
| $v_{13}$ | $(v_{18}, v_{26})$ |
| $v_8$ | $(v_{18}, v_{14})$ |
| $v_5$ | $(v_{34}, v_{47}), (v_{36}, v_{33}), (v_{18}, v_{14})$ |
| $v_3$ | $(v_{20}, v_{14})$ |
| $v_1$ | $(v_2, v_{14})$ |

*Example 2.* Consider again the phylogenetic network in Figure 1. The leaf numbering in Example 1 yields the interval lists listed in Table 2. The network is level-2 and its spread corresponds to the 3 disjoint intervals $(v_{34}, v_{47}), (v_{36}, v_{33}), (v_{18}, v_{14})$ of node $v_5$. □

Now, we consider how to store characteristic vectors under leaf numbering functions having small spread. An efficient approach is to store the starting and ending positions of all intervals in sorted order. We call this representation the *interval list representation* of the clusters. We immediately obtain the following result.

**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.*

*Proof.* For each of the $m$ nodes in $N$, the starting and ending positions of each of its at most $I_f$ intervals are stored in $\lceil 2 \log n \rceil$ bits. □

**Lemma 7.** *Given any leaf numbering function $f$, all descendant leaf bit vectors under $f$ using the interval list representation can be computed in $O(I_f(m+e))$ time.*

*Proof.* Use Algorithm 1 but replace the union operation as follows. Let $v$ be an internal node with children $u$, $w$. Assuming that $C_f[u]$ and $C_f[w]$ are known, $C_f[v]$ can be computed in $O(I_f)$ time by a straightforward algorithm which scans the two sorted position lists for $C_f[u]$ and $C_f[w]$ and merges any intervals which overlap or are neighbors. □

## 4 An Algorithm for Computing the Robinson-Foulds Distance

We now present an algorithm for computing the Robinson-Foulds distance between two input phylogenetic networks $N_1, N_2$ (Algorithm 3).

The algorithm first computes the clusters of $N_1$ and $N_2$ using any of the cluster representations described in the previous sections of this paper. Then, the cardinality of the symmetric difference of the two cluster representations is obtained by radix sorting and simultaneous traversal techniques. Finally, the algorithm outputs the Robinson-Foulds distance between $N_1$ and $N_2$.

**Algorithm 3** *Compute the Robinson-Foulds distance between two phylogenetic networks $N_1, N_2$.*

```
function robinson_foulds_distance(N1, N2)
    cluster_representation(N1, C1); radix sort C1
    cluster_representation(N2, C2); radix sort C2
    m1, m2 ← number of nodes of N1, N2
    i1 ← 1
    i2 ← 1
    c ← 0
    while i1 ⩽ m1 and i2 ⩽ m2 do
        if C1[i1] < C2[i2] then
            i1 ← i1 + 1
        else if C1[i1] > C2[i2] then
            i2 ← i2 + 1
        else
            i1 ← i1 + 1
            i2 ← i2 + 1
            c ← c + 1
    return m1 + m2 − 2 · c
```

**Theorem 1.** *Let $N_1, N_2$ be two phylogenetic networks with $n$ leaves, $m$ nodes, and $e$ edges. The Robinson-Foulds distance between $N_1, N_2$ can be computed in:*

- $O(n(m+e)/\log n)$ *time and* $O(n^2 + n\,m/\log n)$ *words for general networks,*
- $O(n\,m/\log n)$ *time and* $O(n^2 + n\,m/\log n)$ *words for general networks with bounded degree,*

- $O(m + e)$ *time and* $O(m \log n)$ *bits for planar phylogenetic networks,*
- $O((k + 1)(m + e))$ *time and* $O(k\, m \log n)$ *bits for level-*$k$ *phylogenetic networks.*

*Proof.* Implement Algorithm 3 by applying Lemmas 3–7 to obtain the respective cluster representations. The radix sort step and remaining operations can be performed in $O(m\, x)$ time, where $x$ denotes the amount of space needed to represent one cluster. □

## 5 Conclusion

We have presented a new and simple algorithm for computing the Robinson-Foulds distance between two phylogenetic networks. While the fastest known algorithm for computing the Robinson-Foulds distance between two phylogenetic networks with $n$ leaves, $m$ nodes, and $e$ edges runs in $O(m(m + e))$ time, the new algorithm takes advantage of bit-level parallelism and runs in $O(n(m+e)/\log n)$ time on general networks, assuming a word size of $\omega = \lceil \log n \rceil$ bits. In the case of level-$k$ phylogenetic networks, we take advantage of the succinct representation of clusters as intervals of consecutive integers, and the new algorithm runs in $O((k + 1)(m + e))$ time.

We have also introduced a new parameter, the *minimum spread* of a phylogenetic network, and proved that every level-$k$ network has minimum spread at most $k + 1$. For the particular case of bounded-level phylogenetic networks such as planar phylogenetic networks, which include outer-labeled planar split networks and galled-trees, we have shown that the minimum spread is 1, meaning that the new algorithm can be implemented to run in optimal $O(m + e)$ time.

## Acknowledgment

JJ was supported by the Special Coordination Funds for Promoting Science and Technology. TA, RU, GV were supported by the Spanish government and the EU FEDER program under project PCI2006-A7-0603.

## References

1. M. Baroni, C. Semple, and M. Steel. Hybrids in real time. *Syst. Biol.*, 55(1):46–56, 2006.
2. D. Bryant and V. Moulton. Neighbor-Net: An agglomerative method for the construction of phylogenetic networks. *Mol. Biol. Evol.*, 21(2):255–265, 2004.

3. G. Cardona, M. Llabrés, F. Rosselló, and G. Valiente. Metrics for phylogenetic networks I: Generalizations of the Robinson-Foulds metric. *IEEE ACM T. Comput. Biol.*, 6(1):1–16, 2009.
4. G. Cardona, F. Rosselló, and G. Valiente. Comparison of tree-child phylogenetic networks. *IEEE ACM T. Comput. Biol.*, 2009.
5. C. Choy, J. Jansson, K. Sadakane, and W. K. Sung. Computing the maximum agreement of phylogenetic networks. *Theor. Comput. Sci.*, 335(1):93–107, 2005.
6. W. H. E. Day. Optimal algorithms for comparing trees with labeled leaves. *J. Classif.*, 2(1):7–28, 1985.
7. W. F. Doolittle. Phylogenetic classification and the universal tree. *Science*, 284(5423):2124–2128, 1999.
8. S. Grünewald, K. Forslund, A. Dress, and V. Moulton. QNet: An agglomerative method for the construction of phylogenetic networks from weighted quartets. *Mol. Biol. Evol.*, 24(2):532–538, 2007.
9. S. Grünewald, V. Moulton, and A. Spillner. Consistency of the QNet algorithm for generating planar split networks from weighted quartets. *Discr. Appl. Math.*, 157(10):2325–2334, 2009.
10. D. Gusfield, S. Eddhu, and C. Langley. Efficient reconstruction of phylogenetic networks with constrained recombination. In *Proc. 2nd IEEE Computer Society Bioinformatics Conf.*, pages 363–374, 2003.
11. D. Gusfield, S. Eddhu, and C. H. Langley. The fine structure of galls in phylogenetic networks. *INFORMS J. Comput.*, 16(4):459–469, 2004.
12. T. Hagerup. Sorting and searching on the word RAM. In *Proc. 15th Annual Symp. Theoretical Aspects of Computer Science*, volume 1373 of *Lect. Notes Comput. Sci.*, pages 366–398. Springer, 1998.
13. G. Jin, L. Nakhleh, S. Snir, and T. Tuller. Maximum likelihood of phylogenetic networks. *Bioinformatics*, 22(21):2604–2611, 2006.
14. G. Jin, L. Nakhleh, S. Snir, and T. Tuller. Efficient parsimony-based methods for phylogenetic network reconstruction. *Bioinformatics*, 23(2):123–128, 2007.
15. N. D. Pattengale, E. J. Gottlieb, and B. M. Moret. Efficiently computing the Robinson-Foulds metric. *J. Comput. Biol.*, 14(6):724–735, 2007.
16. D. Posada and K. A. Crandall. Intraspecific gene genealogies: Trees grafting into networks. *Trends Ecol. Evol.*, 16(1):37–45, 2001.
17. D. F. Robinson and L. R. Foulds. Comparison of phylogenetic trees. *Math. Biosci.*, 53(1/2):131–147, 1981.
18. F. Rosselló and G. Valiente. All that glisters is not galled. *Math. Biosci.*, 221(1):54–59, 2009.
19. K. Strimmer and V. Moulton. Likelihood analysis of phylogenetic networks using directed graphical models. *Mol. Biol. Evol.*, 17(6):875–881, 2000.
20. K. Strimmer, C. Wiuf, and V. Moulton. Recombination analysis using directed graphical models. *Mol. Biol. Evol.*, 18(1):97–99, 2001.
21. S.-J. Sul, G. Brammer, and T. L. Williams. Efficiently computing arbitrarily-sized Robinson-Foulds distance matrices. In *Proc. 8th Int. Workshop Algorithms in Bioinformatics*, volume 5251 of *Lect. Notes Bioinformatics*, pages 123–134. Springer, 2008.
22. S.-J. Sul and T. L. Williams. An experimental analysis of Robinson-Foulds distance matrix algorithms. In *Proc. 16th Ann. European Symposium on Algorithms*, volume 5193 of *Lect. Notes Comput. Sci.*, pages 793–804. Springer, 2008.
23. L. van Iersel, J. Keijsper, S. Kelk, L. Stougie, F. Hagen, and T. Boekhout. Constructing level-2 phylogenetic networks from triplets. *IEEE ACM T. Comput. Biol.*, 6(4):667–681, 2009.