Algorithms for Molecular Biology BioMed Central

Research

Open Access

Consistency of the Neighbor-Net Algorithm

David Bryant[1], Vincent Moulton*[2] and Andreas Spillner[2]

Address: [1]Department of Mathematics, University of Auckland, Private Bag 92019, Auckland, NZ and [2]School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, UK

Email: David Bryant - bryant@math.auckland.ac.nz; Vincent Moulton* - vincent.moulton@cmp.uea.ac.uk; Andreas Spillner - aspillner@cmp.uea.ac.uk

* Corresponding author

Published: 28 June 2007

Algorithms for Molecular Biology 2007, **2**:8 doi:10.1186/1748-7188-2-8

Received: 26 March 2007
Accepted: 28 June 2007

This article is available from: http://www.almob.org/content/2/1/8

Abstract

Background: Neighbor-Net is a novel method for phylogenetic analysis that is currently being widely used in areas such as virology, bacteriology, and plant evolution. Given an input distance matrix, Neighbor-Net produces a phylogenetic network, a generalization of an evolutionary or phylogenetic tree which allows the graphical representation of conflicting phylogenetic signals.

Results: In general, any network construction method should not depict more conflict than is found in the data, and, when the data is fitted well by a tree, the method should return a network that is close to this tree. In this paper we provide a formal proof that Neighbor-Net satisfies both of these requirements so that, in particular, Neighbor-Net is statistically consistent on circular distances.

1 Background

Phylogenetics is concerned with the construction and analysis of evolutionary or phylogenetic trees and networks to understand the evolution of species, populations and individuals [1]. Neighbor-Net is a phylogenetic analysis and data representation method introduced in [2]. It is loosely based on the popular Neighbor-Joining (NJ) method of Saitou and Nei [3], but with one fundamental difference: whereas NJ constructs phylogenetic trees, Neighbor-Net constructs phylogenetic networks. The method is widely used, in areas such as virology [4], bacteriology [5], plant evolution [6] and even linguistics [7].

Evolutionary processes such as hybridization between species, lateral transfer of genes, recombination within a population, and convergent evolution can all lead to evolutionary histories that are distinctly non tree-like. Moreover, even when the underlying evolution is tree-like, the presence of conflicting or ambiguous signal can make a single tree representation inappropriate. In these situations, phylogenetic network methods can be particularly useful (see e.g. [8]).

Phylogenetic networks are a generalization of phylogenetic trees (see Figure 1 for a typical example of a phylogenetic network). In case there are many conflicting phylogenetic signals supported by the data, Neighbor-Net can represent this conflict graphically. In particular a single network can represent several trees simultaneously, indicate whether or not the data is substantially tree-like, and give evidence for possible reticulation or hybridization events. Evolutionary hypotheses suggested by the network can be tested directly using more detailed phylogenetic analyses and specialized biochemical methods (e.g. DNA fingerprinting or chromosome painting).

For any network construction method, it is vital that the network does not depict more conflict than is found in the

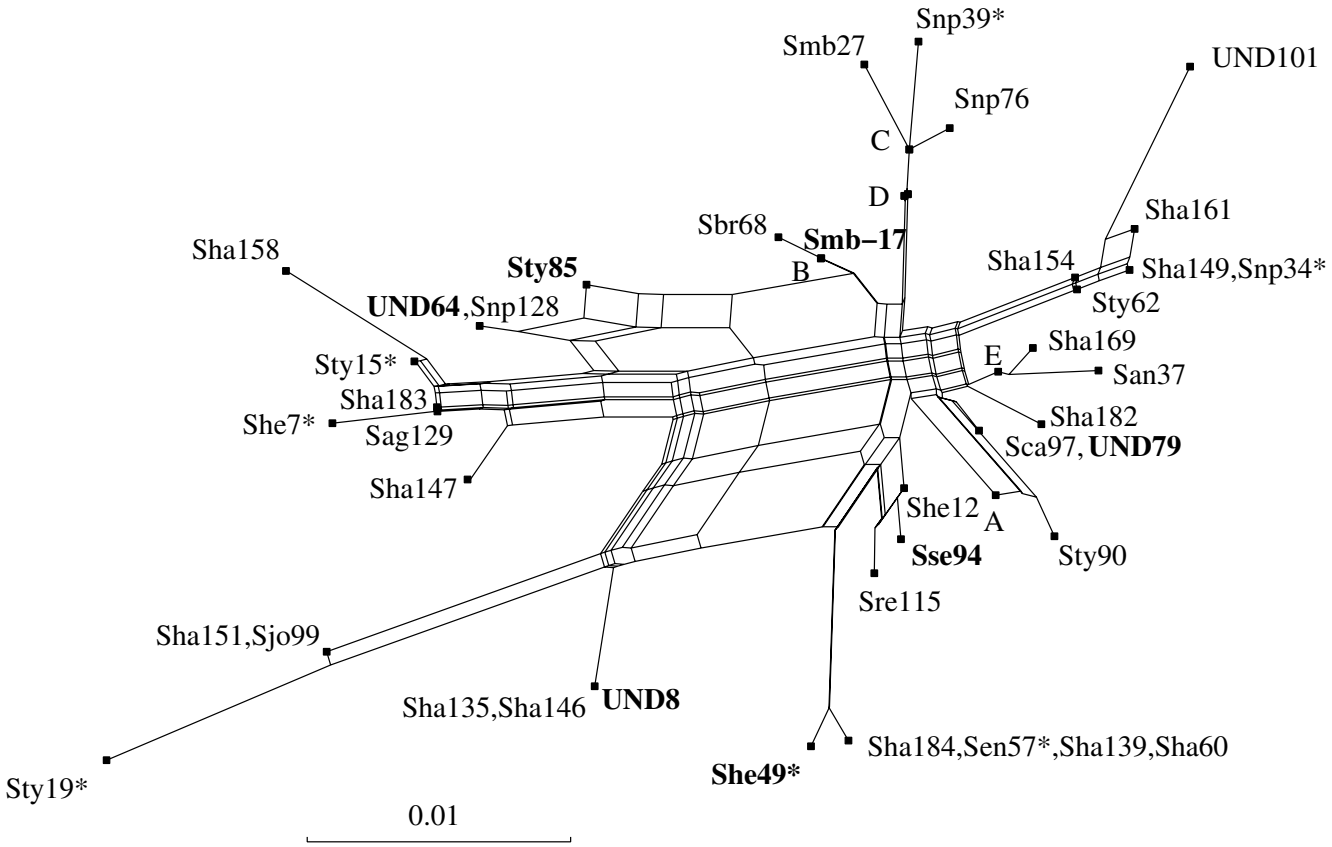

Figure 1

A phylogenetic network. The network was generated by Neighbor-Net for a sequence-based data set comprising of *Salmonella* isolates that originally appeared in [17]. A detailed network-based analysis of this data is presented in [2], where the strains indicated in bold-face are tested for the presence of recombination. Note that the network is planar (that is, it can be drawn in the plane without any crossing edges), and that parallel edges in the network represent bipartitions of the data.

data and that, if there are conflicting signals, then these should be represented by the network. At the same time, when the data is fitted well by a tree, the method should return a network that is close to being a tree. This is essential not just to avoid false inferences, but for the application of networks in statistical tests of the extent to which the data is tree-like [9].

In this paper we provide a proof that these properties all hold for Neighbor-Net. Formally, we prove that if the input to NeighborNet is a circular distance function (distance matrix) [10], then the method returns a network that exactly represents the distance. Circular distance functions are more general than additive (patristic) distances on trees and, thus, as a corollary, if Neighbor-Net is given an additive distance it will return the corresponding tree. In this sense, Neighbor-Net is a statistically consistent method.

The paper is structured as follows: In Section 2 we introduce some basic notation, and in Section 3 we review the Neighbor-Net algorithm. In Section 4 we prove that Neighbor-Net is consistent (Theorem 4.1).

2 Preliminaries

In this section we present some notation that will be needed to describe the Neighbor-Net algorithm. We will assume some basic facts concerning phylogenetic trees, more details concerning which may be found in [11].

Throughout this paper, X will denote a finite set with cardinality n. A *split* $S = \{A, B\}$ (of X) is a bipartition of X. We let $\Sigma = \Sigma(X) = \{\{A, X\backslash A\} | \emptyset \subset A \subset X\}$ denote the set of all splits of X, and call any non-empty subset of $\Sigma(X)$ a *split system*. A *split weight function on* X is a map $\omega: \Sigma(X) \to \mathbb{R}_{\geq 0}$. We let Σ_ω denote the set $\{S \in \Sigma | \omega(S) > 0\}$, the *support* of ω.

Let $\Theta = x_1, \ldots, x_n$ be an ordering of X. A split $S = \{A, B\}$ is *compatible with* Θ if there exist $i, j \in \{1, \ldots, n\}$, $i \leq j$, such that $A = \{x_i, \ldots, x_j\}$ or $B = \{x_i, \ldots, x_j\}$. Note that if a split is compatible with an ordering Θ it is also compatible with its reversal $x_n, \ldots, x_2, x_1$ and with ordering $x_2, \ldots, x_n, x_1$. We

let $_\Theta$ denote the set of those splits in (X) which are compatible with ordering Θ. A split system $'$ is *compatible with* Θ if $' \subseteq _\Theta$. In addition a split system $' \subseteq (X)$ is *circular* if there exists an ordering Θ of X such that $'$ is compatible with Θ. Note that any split system corresponding to a phylogenetic tree is circular [[11], Ch. 3], and so circular split systems can be regarded as a generalization of split systems induced by phylogenetic trees. A split weight function ω is called *circular* if the split system $_\omega$ is circular. A *distance function on* X is a map $d: X \times X \to \mathbb{R}_{\geq 0}$ such that for all $x, y \in X$ both $d(x, x) = 0$ and $d(x, y) = d(y, x)$ hold. Note that any split weight function ω on X induces a distance function d_ω on X as follows: For a split $S = \{A, B\} \in (X)$ define the distance function or *split metric* d_S by

$$d_S(x,y) = \begin{cases} 0 & \text{if } \{x,y\} \subseteq A \text{ or } \{x,y\} \subseteq B \\ 1 & \text{otherwise,} \end{cases}$$

and put

$$d_\omega(x,y) = \sum_{S \in \mathfrak{S}(X)} \omega(S) d_S(x,y)$$

for all $x, y \in X$. A distance function d is called *circular* if there exits a circular split weight function ω such that $d = d_\omega$. An ordering Θ of X is said to be compatible with d if there exists ω such that $d = d_\omega$ and $_\omega \subseteq _\Theta$. Note that the representation of a circular distance function d is unique, i.e., if $d = d_{\omega_1}$ and $d = d_{\omega_2}$ for circular split weight functions ω_1 and ω_2 then $\omega_1 = \omega_2$ holds [10].

Circular distances were introduced in [10] and have been further studied in, for example, [12] and [13]. Just as any tree-like distance function on X can be uniquely represented by a phylogenetic tree [[11], ch. 7], any circular distance function d can be represented by a planar phylogenetic network such as the one pictured in Figure 1[14]. The program SplitsTree [9] allows the automatic generation of such a network for d by computing a circular split weight function ω with $d = d_\omega$.

3 Description of the Neighbor-Net algorithm

In this section we present a detailed description of the Neighbor-Net algorithm, as implemented in the current version of SplitsTree [9]. The Neighbor-Net algorithm was originally described in [2], where the reader may find a more informal description for how it works. For the convenience of the reader we will use the same notation as in [2] where possible.

In Figure 2 we present pseudo-code for the Neighbor-Net algorithm. The aim of the algorithm is, for a given input distance function d, to compute a circular split weight function ω so that the distance function d_ω gives a good approximation to d. The resulting distance function d_ω can then be represented by a planar phylogenetic network as indicated in the last section.

To this end, NEIGHBOR-NET first computes an ordering Θ of X, and then applies a non-negative least-squares procedure to find a best fit for d within the set of distance functions $\{d_\varphi | \varphi: (X) \to \mathbb{R}_{\geq 0}, _\varphi \subseteq _\Theta\}$. More details concerning the least-squares procedure may be found in [2]: Here we will concentrate on the description of the key computation for finding an ordering Θ of X, which is detailed in the procedure FINDORDERING.

An (*ordered*) *cluster* is a non-empty finite set C together with an ordering $\Theta_C = c_1, \ldots, c_k$ of the elements in C, $k = |C|$. Two elements $a, b \in C$ are called *neighbors* if there exists $i \in \{1, \ldots, k - 1\}$ such that $a = c_i$ and $b = c_{i+1}$, or $b = c_i$ and $a = c_{i+1}$. The input of the procedure FINDORDERING consists of a set $\mathfrak{C}$ of mutually disjoint clusters, together with a distance function d on the set $Y = \bigcup_{C \in \mathfrak{C}} C$. The ordering $\Theta = y_1, \ldots, y_n$ of Y that is returned by FINDORDERING must be *compatible* with the collection $\mathfrak{C}$ of ordered clusters, that is, for every cluster $C \in \mathfrak{C}$ there must exist $i, j \in \{1, \ldots, n\}$, $i \leq j$, with the property that $\Theta_C = y_i, \ldots, y_j$ or $\Theta_C = y_j, \ldots, y_i$.

The procedure FINDORDERING calls itself recursively. Apart from the base case (line 5 of Figure 2), where the recursion bottoms out, two different cases are considered – the *reduction* and *selection* cases (lines 7–15 and lines 17–22 of Figure 2, respectively). In the reduction case a cluster $C \in \mathfrak{C}$ with $k = |C| \geq 3$ is replaced by a smaller cluster C'. In particular, in lines 7–11 we let $\Theta_C = c_1, \ldots, c_k$ be the ordering of C with $c_1 = x$, $c_2 = y$, $c_3 = z$, and put $C' = (C \setminus \{x, y, z\}) \cup \{u, v\}$ and $\Theta_{C'} = u, v, c_4, \ldots, c_k$, where u and v are two new elements not contained in Y. Then, in lines 12–14, we define a distance function d' on the set $Y' = (Y \setminus \{x, y, z\}) \cup \{u, v\}$ using the formulae:

$$\begin{aligned} d'(a,b) &= d(a,b) && \text{for } \{a,b\} \subseteq Y' \setminus \{u,v\} \\ d'(u,a) &= (\alpha + \beta) d(x,a) + \gamma d(y,a) && \text{for } a \in Y' \setminus \{u,v\} \\ d'(v,a) &= \alpha d(y,a) + (\beta + \gamma) d(z,a) && \text{for } a \in Y' \setminus \{u,v\} \\ d'(u,v) &= \alpha d(x,y) + \beta d(x,z) + \gamma d(y,z) \end{aligned} \quad (1)$$

where α, β and γ are positive real numbers satisfying $\alpha + \beta + \gamma = 1$ (note that these formulae slightly differ from the ones given in [2] in which there is a typographical error).

NEIGHBOR-NET(X, d)

Input: A finite non-empty set X and a distance function d on X
Output: A circular split weight function ω

1. $\mathfrak{C} = \{\{x\} \mid x \in X\}$ //initial set of clusters
2. Θ = FINDORDERING($\mathfrak{C}$, d)
3. ω = ESTIMATESPLITWEIGHTS(X, d, Θ)
4. **return** ω

FINDORDERING($\mathfrak{C}$, d)

Input: A collection $\mathfrak{C}$ of ordered clusters and a distance function d
Output: An ordering Θ of the elements in $\cup_{C \in \mathfrak{C}} C$

1. $Y = \cup_{C \in \mathfrak{C}} C$
2. $m = |\mathfrak{C}|$
3. $n = |Y|$
4. **if** $n \leq 3$ //base case
5. **return** an ordering Θ of Y that is compatible with $\mathfrak{C}$.
6. **else if** there exists $C \in \mathfrak{C}$ with $k = |C| \geq 3$ //reduction case
7. Select $x = c_1$, $y = c_2$ and $z = c_3$ from C with $\Theta_C = c_1, \ldots, c_k$.
8. Create two new elements u, v not contained in Y.
9. $C' = (C \setminus \{x, y, z\}) \cup \{u, v\}$
10. $\Theta_{C'} = u, v, c_4, \ldots, c_k$
11. $\mathfrak{C}' = (\mathfrak{C} \setminus \{C\}) \cup \{C'\}$
12. Compute distance function d' on $Y' = \cup_{C \in \mathfrak{C}'} C$ according to (1).
13. Θ' = FINDORDERING($\mathfrak{C}'$, d')
14. Compute an ordering Θ of Y according to (2).
15. **return** Θ
16. **else** //selection case
17. Select two clusters $C_1, C_2 \in \mathfrak{C}$ that minimize (3).
18. $C' = C_1 \cup C_2$
19. Compute ordering $\Theta_{C'}$ using (4).
20. $\mathfrak{C}' = (\mathfrak{C} \setminus \{C_1, C_2\}) \cup \{C'\}$
21. Θ = FINDORDERING($\mathfrak{C}'$, d)
22. **return** Θ

Figure 2
The Neighbor-Net algorithm. Pseudo-code for the Neighbor-Net algorithm detailing the procedure FINDORDERING.

In the current implementation of Neighbor-Net the values $\alpha = \beta = \gamma = 1/3$ are used.

When FINDORDERING is recursively called with the new collection $\mathfrak{C}'$ of clusters and distance function d' it returns an ordering $\Theta' = y'_1,...,y'_{n-1}$ of Y' that is compatible with $\mathfrak{C}'$. Thus, there exists $i \in \{1, ..., n - 2\}$ such that either $u = y'_i$ and $v = y'_{i+1}$ or $v = y'_i$ and $u = y'_{i+1}$. The resulting ordering Θ of Y is then defined (in line 14) as follows:

$$\Theta = \begin{cases} y'_1,...,y'_{i-1},x,y,z,y'_{i+2},...,y'_{n-1} & \text{if } u = y'_i \text{ and } v = y'_{i+1} \\ y'_1,...,y'_{i-1},z,y,x,y'_{i+2},...,y'_{n-1} & \text{if } u = y'_{i+1} \text{ and } v = y'_i. \end{cases} \quad (2)$$

This completes the description of the reduction case.

We now describe the selection case. Note that in view of line 6 this case only applies if every cluster in $\mathfrak{C}$ contains at most two elements. In lines 17–18, two clusters $C_1, C_2 \in \mathfrak{C}$ are selected and replaced by the single cluster $C' = C_1 \cup C_2$. The clusters C_1 and C_2 are selected as follows: We define a distance function $\bar{d}$ on the set of clusters $\mathfrak{C}$ by

$$\bar{d}(A,B) = \begin{cases} 0 & \text{if } A = B \\ \dfrac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B} d(a,b) & \text{if } A \neq B, \end{cases}$$

and select $C_1, C_2 \in \mathfrak{C}$, $C_1 \neq C_2$ that minimize the quantity

$$Q(C_1,C_2) = (m-2)\bar{d}(C_1,C_2) - \sum_{C\in\mathfrak{C}\setminus\{C_1\}} \bar{d}(C_1,C) - \sum_{C\in\mathfrak{C}\setminus\{C_2\}} \bar{d}(C_2,C) \quad (3)$$

where m is the number of clusters in $\mathfrak{C}$. The function Q that is used to select pairs of clusters is called the *Q-criterion*. Note that this is a direct generalization of the selection criterion used in the NJ algorithm [2]. However, using only this criterion yields a method that is not consistent as illustrated in Figure 3. So, once the clusters C_1 and C_2 have been selected we use a second criterion to determine an ordering $\Theta_{C'}$ in line 19 for the new cluster C'. In particular, for every $x \in C_1 \cup C_2$ we define

$$R(x) = \sum_{C\in\mathfrak{C}\setminus\{C_1,C_2\}} \bar{d}(\{x\},C) + \sum_{y\in(C_1\cup C_2)\setminus\{x\}} d(x,y),$$

put $\hat{m} = m + |C_1| + |C_2| - 2$, and select $x_1 \in C_1$ and $x_2 \in C_2$ that minimize the quantity

$$\hat{Q}[d](x_1, x_2) = (\hat{m} - 2)d(x_1, x_2) - R(x_1) - R(x_2). \quad (4)$$

We then choose an ordering $\Theta_{C'}$ in which x_1 and x_2 are neighbors and for which every two elements that were neighbors in C_1 or C_2 remain neighbors. This completes the description of the selection case, and hence the description of the procedure FINDORDERING.

4 Neighbor-Net is consistent

In this section we prove the consistency of Neighbor-Net:

Theorem 4.1 If $d: X \times X \to \mathbb{R}_{\geq 0}$ is a circular distance function, then the output of the Neighbor-Net algorithm is a circular split weight function $\omega: (X) \to \mathbb{R}_{\geq 0}$ with the property that $d = d_\omega$.

The key part of the Neighbor-Net algorithm is the procedure FINDORDERING. We will show that, for a circular distance function $d = d_\omega$ on X, the call FINDORDERING($\{\{x\}|x \in X\}, d$) will produce an ordering Θ of X that is compatible with d. The non-negative least squares procedure finds the distance function in $\{d_\varphi|\varphi: (X) \to \mathbb{R}_{\geq 0}, \varphi \subseteq_\Theta\}$ that is closest to d. As this set of distance functions includes d_ω, the least squares procedure returns exactly $d = d_\omega$, proving the theorem.

We focus, then, on the proof that FINDORDERING behaves as required:

Theorem 4.2 Let $d: Y \times Y \to \mathbb{R}_{\geq 0}$ be a distance function that is induced by a circular split weight function $\omega: (Y) \to \mathbb{R}_{\geq 0}$. In addition, let $\mathfrak{C}$ be a collection of mutually disjoint clusters with the property that $Y = Y = \bigcup_{C\in\mathfrak{C}} C$, and assume there exists an ordering of Y that is compatible with ω and with $\mathfrak{C}$. Then FINDORDERING($\mathfrak{C}$, d) will compute an ordering that is compatible with the collection of clusters $\mathfrak{C}$ and with the split weight function ω.

We present the proof of this result in the remainder of this section. Suppose that the algorithm FINDORDERING is called with input $\mathfrak{C}$ and d and that there exists an ordering that is compatible with $\mathfrak{C}$ and d. Let $Y = \bigcup_{C\in\mathfrak{C}} C$. We prove Theorem 4.2 by induction, first on $|Y|$, the cardinality of Y, and then on $|\mathfrak{C}|$, the number of clusters in $\mathfrak{C}$.

The *base case* of the induction is $|Y| \leq 3$. In this case the set of splits $_\Theta$ equals (Y) for every ordering of Y. In particular,

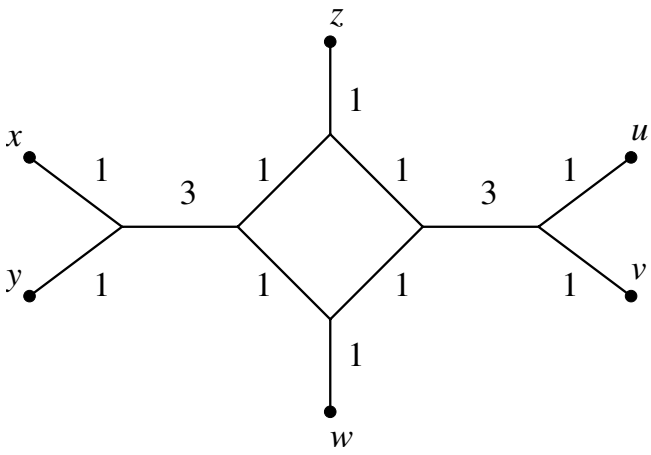

Figure 3
A network representing a circular distance. A circular distance d on the set {u, v, ..., z} for which NeighborNet using only the Q-criterion employed in NJ to cluster elements would be inconsistent. Distances are given by shortest paths in the network. The pairs u, v and x, y would be clustered together first and then the pair z, w. However it is not hard to show that z and w are not adjacent in any ordering of {u, v, ..., z} that is compatible with d.

any ordering of Y that is compatible with $\mathfrak{C}$ is also compatible with ω.

We now assume that $|Y| > 3$ and make the following *induction hypothesis*:

> If there exists an ordering compatible with distance function d' and ordered clusters $\mathfrak{C}'$, where either $|\bigcup_{C\in\mathfrak{C}'}C| < |Y|$, or $|\bigcup_{C\in\mathfrak{C}'}C| = |Y|$ and $|\mathfrak{C}'| < |\mathfrak{C}|$, then FINDORDERING($\mathfrak{C}'$, d') will return an ordering compatible with $\mathfrak{C}'$ and d'.

There are two cases to consider. In the first case, $\mathfrak{C}$ contains some cluster C with $|C| \geq 3$. In the second case, $\mathfrak{C}$ contains only clusters C with $|C| \leq 2$.

4.1 Case 1: The reduction case

Suppose that there is $C \in \mathfrak{C}$ with $|C| \geq 3$. This is the *reduction case* in the description of the algorithm. The procedure FINDORDERING constructs a new set of clusters $\mathfrak{C}'$ (in line 11) and a new distance function d' (in line 12). We first show that, if there is an ordering compatible with $\mathfrak{C}$ and d, then there is also an ordering compatible with $\mathfrak{C}'$ and d'.

Proposition 4.3 If $\mathfrak{C}'$ and d' are constructed according to lines 7–12 of the procedure FINDORDERING then there exists an ordering compatible with $\mathfrak{C}'$ and d'.

Proof: Suppose that $\tilde{\Theta} = \gamma_1, ..., \gamma_n$ is an ordering of Y that is compatible with $\mathfrak{C}$ and d, where, without loss of generality, we have $\Theta_C = \gamma_1, ..., \gamma_k$. Let $\tilde{\Theta}' = u, v, \gamma_4, ..., \gamma_n = z_1, ..., z_{n-1}$, which is an ordering of $Y' = \bigcup_{C\in\mathfrak{C}'}C$. We claim that the ordering $\tilde{\Theta}'$ is compatible with the collection $\mathfrak{C}'$ and with the distance function d'.

Since $\mathfrak{C}$ is compatible with $\tilde{\Theta}$ it is straight-forward to check that $\mathfrak{C}'$ is compatible with $\tilde{\Theta}'$. Hence, we only need to show that $\tilde{\Theta}'$ is compatible with d'. We will use a 4-point condition that was first studied in a different context by Kalmanson [15] and has been shown to characterize circular distances in [12]. To be more precise, it suffices to show that, for every four elements $z_{i_1}, z_{i_2}, z_{i_3}, z_{i_4}$, $i_1 < i_2 < i_3 < i_4$,

$$d'(z_{i_1}, z_{i_3}) + d'(z_{i_2}, z_{i_4}) \geq d'(z_{i_1}, z_{i_2}) + d'(z_{i_3}, z_{i_4}) \text{ and}$$
$$d'(z_{i_1}, z_{i_3}) + d'(z_{i_2}, z_{i_4}) \geq d'(z_{i_1}, z_{i_4}) + d'(z_{i_2}, z_{i_3}).$$

Case 1: $|\{z_{i_1}, z_{i_2}, z_{i_3}, z_{i_4}\} \cap \{u, v\}| = 0$. The above inequalities follow immediately since d is circular, and d and d' as well as $\tilde{\Theta}$ and $\tilde{\Theta}'$ coincide on $Y'\backslash\{u, v\}$.

Case 2: $|\{z_{i_1}, z_{i_2}, z_{i_3}, z_{i_4}\} \cap \{u, v\}| = 1$. Consider the situation $z_{i_1} = u$. Then

$$\begin{aligned} &d'(z_{i_1}, z_{i_3}) + d'(z_{i_2}, z_{i_4}) \\ &= (\alpha+\beta)d(x, z_{i_3}) + \gamma d(y, z_{i_3}) + (\alpha+\beta+\gamma)d(z_{i_2}, z_{i_4}) \\ &\geq (\alpha+\beta)d(x, z_{i_2}) + \gamma d(y, z_{i_2}) + (\alpha+\beta+\gamma)d(z_{i_3}, z_{i_4}) \\ &= d'(z_{i_1}, z_{i_2}) + d'(z_{i_3}, z_{i_4}). \end{aligned}$$

The other inequalities can be derived in a completely analogous way.

Case 3: $|\{z_{i_1}, z_{i_2}, z_{i_3}, z_{i_4}\} \cap \{u, v\}| = 2$. Then we have $z_{i_1} = u$ and $z_{i_2} = v$ and

$$
\begin{aligned}
&d'(z_{i_1}, z_{i_3}) + d'(z_{i_2}, z_{i_4}) \\
&\quad = (\alpha+\beta)d(x, z_{i_3}) + \gamma d(y, z_{i_3}) + \alpha d(y, z_{i_4}) + (\beta+\gamma)d(z, z_{i_4}) \\
&\quad \geq \alpha d(x, y) + \beta d(x, z) + \gamma d(y, z) + (\alpha+\beta+\gamma)d(z_{i_3}, z_{i_4}) \\
&\quad = d'(z_{i_1}, z_{i_2}) + d'(z_{i_3}, z_{i_4}).
\end{aligned}
$$

The other inequality $d'(z_{i_1}, z_{i_3}) + d'(z_{i_2}, z_{i_4}) \geq d'(z_{i_1}, z_{i_4}) + d'(z_{i_2}, z_{i_3})$ can be shown to hold in a similar way. ■

The procedure FINDORDERING calls itself recursively with $\mathfrak{C}'$ and d' as input. An ordering of Y', the union of $\mathfrak{C}'$, is returned. By Proposition 4.3 and the induction hypothesis, this ordering Θ' is compatible with $\mathfrak{C}'$ and d'. It is used to construct an ordering Θ on Y, in line 14, which becomes the output of the procedure.

Proposition 4.4 The ordering Θ is compatible with collection $\mathfrak{C}$ and with the distance function d.

Proof: Since $\mathfrak{C}'$ is compatible with Θ' it is straight-forward to check that $\mathfrak{C}$ is compatible with Θ. Hence we only need to show that Θ is compatible with d.

Let orderings $\tilde{\Theta} = y_1, \ldots, y_n$ of Y and $\tilde{\Theta}' = z_1, \ldots, z_{n-1}$ of Y' be as in the proof of Proposition 4.3 and let ω be the split weight function such that $d = d_\omega$. Then $\tilde{\Theta}$ is compatible with all splits S such that $\omega(S) > 0$. Now consider some split $S = \{A, B\}$ such that $\omega(S) > 0$ and assume that $y_n \in B$. Then there exists $i, j \in \{1, \ldots, n-1\}$, $i \leq j$, such that $A = \{y_i, \ldots, y_j\}$. Note also that, since the distance function d' is compatible with ordering $\tilde{\Theta}' = z_1, \ldots, z_{n-1}$ of Y' and, hence, is circular, there exists a unique circular split weight function ω': $(Y') \to \mathbb{R}_{\geq 0}$ with the property that $d' = d_{\omega'}$. We divide the remaining argument into five cases.

Case 1: $j \leq 3$. Then, clearly, S is compatible with Θ.

Case 2: $j \geq 4$ and $i = 1$. Define $A' = \{z_1, \ldots, z_{j-1}\}$ and the split $S' = \{A', Y'\backslash A'\}$ of Y'. Then we can express $\omega'(S')$ in terms of d' as follows (cf. [12]):

$$
\begin{aligned}
2\omega'(S') &= d'(z_1, z_j) + d'(z_{j-1}, z_{n-1}) - d'(z_1, z_{j-1}) - d'(z_j, z_{n-1}) \\
&= (\alpha+\beta)d(y_1, y_{j+1}) + \gamma d(y_2, y_{j+1}) + d(y_j, y_n) \\
&\quad -(\alpha+\beta)d(y_1, y_j) - \gamma d(y_2, y_j) - d(y_{j+1}, y_n) \\
&\geq (\alpha+\beta+\gamma)(d(y_1, y_{j+1}) + d(y_j, y_n) - d(y_1, y_j) - d(y_{j+1}, y_n)) \\
&= 2\omega(S)
\end{aligned}
$$

Thus, $\omega'(S') > 0$. Hence, the split S' is compatible with the ordering Θ' of Y'. But then the split S is compatible with the ordering Θ of Y.

Case 3: $j \geq 4$ and $2 \leq i \leq 3$. We only consider the situation when $i = 2$; the situation $i = 3$ is completely analogous. Define $A' = \{z_2, \ldots, z_{j-1}\}$ and the split $S' = \{A', Y'\backslash A'\}$ of Y'. With a similar calculation as made for Case 2 we obtain $\omega'(S') \geq (\alpha+\beta)\omega(S)$. Hence, $\omega'(S') > 0$ and, thus, S' is compatible with Θ'. But then S is compatible with Θ.

Case 4: $j \geq 4$ and $i = 4$. This case is similar to Case 2. Define $A' = \{z_4, \ldots, z_{j-1}\}$ and $S' = \{A', Y'\backslash A'\}$. We obtain $\omega'(S') \geq \omega(S)$. Hence, as for Case 2, $\omega'(S') > 0$ and, thus, S is compatible with Θ.

Case 5: $j \geq i \geq 5$. Define the split $S' = \{A, Y'\backslash A\}$. Then we have $\omega'(S') = \omega'(S') > 0$. Hence, S' is compatible with Θ' and, thus, S is compatible with Θ. ■

4.2 Case 2: The selection case

Now suppose that there are no clusters $C \in \mathfrak{C}$ with $|C| \geq 3$. This is the *selection case* in the description of the algorithm.

In line 17 the algorithm selects two clusters that minimize (3):

$$
Q(C_1, C_2) = (m-2)\bar{d}(C_1, C_2) - \sum_{C \in \mathfrak{C}\backslash\{C_1\}} \bar{d}(C_1, C) - \sum_{C \in \mathfrak{C}\backslash\{C_2\}} \bar{d}(C_2, C),
$$

where

$$
\bar{d}(A, B) = \begin{cases} 0 & \text{if } A = B \\ \dfrac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a, b) & \text{if } A \neq B. \end{cases}
$$

Note that $\bar{d}$ is a distance function defined on the set of clusters $\mathfrak{C}$. We will first show that $\bar{d}$ is circular. We do this in two steps: Proposition 4.5 and Proposition 4.6.

Proposition 4.5 Let d: $M \times M \to \mathbb{R}_{\geq 0}$ be a circular distance function and $\Theta = x_1, \ldots, x_n$ be an ordering of M that is compatible with d. Let $M' = (M\backslash\{x_1, x_2\}) \cup \{\gamma\}$ where γ is a

new element not contained in M. Define a distance function d': $M' \times M' \to \mathbb{R}_{\geq 0}$ as follows:

$$d'(a,b) = d(a,b) \quad \text{for } \{a,b\} \subseteq M' \setminus \{\gamma\}$$
$$d'(\gamma,a) = \lambda d(x_1,a) + (1-\lambda)d(x_2,a) \quad \text{for } a \in M' \setminus \{\gamma\},$$

where λ is a real number with the property that $0 < \lambda < 1$. Then the following hold:

(i) d' is circular and compatible with ordering $\gamma, x_3, \ldots, x_n$ of M'.

(ii) If $z_1, \ldots, z_{n-1}$ is an ordering of M' that is compatible with d' then at least one of the orderings $x_1, x_2, z_2, \ldots, z_{n-1}$ or $x_2, x_1, z_2, \ldots, z_{n-1}$ of M is compatible with d.

Proof: (i) and (ii) can be proven using convexity arguments, or in a way analogous to our proof of Propositions 4.3 and 4.4, respectively. ■

Proposition 4.6 The distance function $\bar{d}$, defined on the individual clusters in $\mathfrak{C}$, is a circular distance. Moreover, for every ordering $D_1, \ldots, D_k$ of $\mathfrak{C}$ that is compatible with $\bar{d}$ there exist orderings Θ_i of D_i, $i \in \{1, \ldots, k\}$, such that the ordering $\Theta_1, \ldots, \Theta_k$ of Y is compatible with distance function d.

Proof: We use multiple applications of Proposition 4.5, once for each cluster in $\mathfrak{C}$ with two elements, and with $\lambda = \frac{1}{2}$ in each case. ■

We now have the more difficult task of showing that clusters C_1 and C_2 selected by the Q-criterion, that is by minimizing (3), are adjacent in at least one ordering of the clusters that is compatible with $\bar{d}$, as described in Proposition 4.6. This is the most technical part of the proof. The key step is the inequality established in Lemma 4.7. This is used to prove Theorem 4.8, which establishes that the Q-criterion when applied to a circular distance will always select a pair of elements that are adjacent in at least one ordering compatible with the circular distance. As a corollary it will follow that there exists an ordering of the clusters in $\mathfrak{C}$ compatible with $\bar{d}$ where C_1 and C_2 are adjacent.

Lemma 4.7 Let $\Theta = x_1, x_2, \ldots, x_n$ be an ordering of M that is compatible with circular distance d on M and suppose that $3 \leq r \leq \lceil n/2 \rceil$. Let $S = \{A, M \backslash A\}$ be a split compatible with Θ where $A = \{x_i, \ldots, x_j\}$. Define Q_S: $M \times M \to \mathbb{R}$ by

$$Q_S(x_i,x_j) = (n-2)d_S(x_i,x_j) - \sum_{k=1}^{n} d_S(x_i,x_k) - \sum_{k=1}^{n} d_S(x_j,x_k)$$

and let

$$\lambda(S) = \sum_{l=1}^{r-1} Q_S(x_l,x_{l+1}) - (r-1)Q_S(x_1,x_r).$$

(i) If $\min\{|A|, |M \backslash A|\} > 1$ and $|A \cap \{x_1, x_r\}| = 1$ then $\lambda(S) < 0$.

(ii) Any other split S compatible with Θ satisfies $\lambda(S) \leq 0$.

Proof: Expanding $\lambda(S)$ gives

$$\begin{aligned}\lambda(S) = (n-2)\sum_{l=1}^{r-1} d_S(x_l,x_{l+1}) \quad & -(r-1)(n-2)d_S(x_1,x_r) \\ +(r-2)\sum_{i=1}^{n} d_S(x_1,x_l) \quad & -2\sum_{l=2}^{r-1}\sum_{k=1}^{n} d_S(x_l,x_k) \\ +(r-2)\sum_{l=1}^{n} d_S(x_r,x_l).\end{aligned}$$

We divide the rest of our argument into five cases which are summarized in Table 1. For these cases straight-forward calculations yield the entries of Table 2. Using Table 2 we compute $\lambda(S)$ in each case.

Case (i): We obtain $\lambda(S) = 2(j-1)(j+1-r) + 2(j-1)(j+1-n)$. Hence, $\lambda(S) = 0$ if $j = 1$ and $\lambda(S) < 0$ if $j \geq 2$.

Case (ii): We obtain $\lambda(S) = 0$.

Case (iii): We obtain $\lambda(S) = (j-i)(4(j-i) - 2n + 8)$. Thus, since $j - i \leq r - 3 \leq (n+1)/2 - 3$, $\lambda(S) = 0$ if $i = j$ and $\lambda(S) < 0$ if $i < j$.

Case (iv): We obtain $\lambda(S) = 2(i-r)(n-2-(j-i)) + 2(2-i)(j-i)$. Thus, since $j - i \leq n - 3$, $\lambda(S) < 0$ if $i < r$. If $i = r$ then $\lambda(S) = 0$ if $j = r$ and $\lambda(S) < 0$ otherwise.

Case (v): We obtain $\lambda(S) = 0$. ■

Theorem 4.8 Let M be a set of n elements and d: $M \times M \to \mathbb{R}_{\geq 0}$ be a circular distance function. Suppose that x, y minimize

$$Q(x,y) = (n-2)d(x,y) - \sum_{z \in M} d(x,z) - \sum_{z \in M} d(y,z).$$

Then there is an ordering of M that is compatible with d in which x and y are adjacent.

Proof: Let $\Theta = x_1, \ldots, x_n$ be an ordering of M that is compatible with d. Suppose that $Q(x_1, x_r) \leq Q(x, y)$ for all x, y where, without loss of generality, $2 \leq r \leq \lceil n/2 \rceil$. If $r = 2$ then we are done, so we assume $r \geq 3$. Let ω be the (circular) split weight function for which $d = d_{\omega}$, so Θ is compatible with ω. Let Θ^* be the ordering obtained by removing x_r from Θ and re-inserting it immediately after x_1. We claim that Θ^* is also compatible with ω.

As in Lemma 4.7, for any split S compatible with Θ we define

$$\lambda(S) = \sum_{l=1}^{r-1} Q_S(x_l, x_{l+1}) - (r-1)Q_S(x_1, x_r).$$

By the choice of x_1 and x_r we have

$$(r-1)Q(x_1, x_r) \leq \sum_{l=1}^{r-1} Q(x_l, x_{l+1}).$$

Since Q is linear, and $d = \Sigma_{S \in (X)} \omega(S) d_S$ by Lemma 4.7 we have

$$\begin{aligned} 0 &\leq \sum_{l=1}^{r-1} Q(x_l, x_{l+1}) - (r-1)Q(x_1, x_r) \\ &= \sum_S \omega(S) \left(\sum_{l=1}^{r-1} Q_S(x_l, x_{l+1}) - (r-1)Q_S(x_1, x_r) \right) \\ &= \sum_S \omega(S)\lambda(S) \leq 0. \end{aligned}$$

Now consider any split S compatible with Θ but not Θ^*. Then S satisfies the conditions in Lemma 4.7 (i), giving $\lambda(S) < 0$ and hence $\omega(S) = 0$. Thus there are no splits in the support of ω that are not compatible with Θ^*, and Θ^* is compatible with ω and, hence, d. Thus x_1 and x_r are adjacent in an ordering Θ^* compatible with d. ■

Corollary 4.9 Let C_1 and C_2 be the two clusters selected in line 17 of procedure FINDORDERING. Then there exists an ordering $\Theta^* = D_1, \ldots, D_k$ of $\mathfrak{C}$ such that $D_1 = C_1$, $D_2 = C_2$ and $\overline{d}$ is compatible with Θ^*.

After selecting C_1 and C_2 the procedure FINDORDERING removes these clusters from the collection and replaces them with their union $C' = C_1 \cup C_2$. It also assigns an ordering $\Theta_{C'}$ to the cluster.

FINDORDERING is then called recursively. The following is directly analogous to Proposition 4.3.

Proposition 4.10 There exists an ordering of Y that is compatible with collection $\mathfrak{C}'$ and split weight function ω.

Proof: We already know by Proposition 4.9 and Proposition 4.6 that there exists an ordering $\tilde{\Theta} = y_1, \ldots, y_n$ of Y that is compatible with $\mathfrak{C}$ and ω and, in addition, also satisfies one of the following properties:

$C_1 = \{y_1\}$ and $C_2 = \{y_2\}$ $\quad$ $C_1 = \{y_1\}$ and $C_2 = \{y_2, y_3\}$
$C_1 = \{y_1, y_2\}$ and $C_2 = \{y_3\}$ $\quad$ $C_1 = \{y_1, y_2\}$ and $C_2\{y_3, y_4\}$.

If $x_1 \in C_1$ and $x_2 \in C_2$ are selected such that $\tilde{\Theta}$ is also compatible with $\mathfrak{C}'$ then we are done. Otherwise we have to construct a suitable new ordering $\tilde{\Theta}'$ of Y. There are, up to symmetric situations with roles of C_1 and C_2 swapped, only two cases we need to consider.

Case 1: $C_1 = \{y_1, y_2\}$, $x_1 = y_1$ and $x_2 = y_3$. We want to show that ordering $\tilde{\Theta}' = y_2, y_1, y_3, \ldots, y_n$ is compatible with ω. To this end we first show that $\hat{Q}[d](y_2, y_3) \leq \hat{Q}[d](y_1, y_3)$. It suffices to establish this inequality for all split metrics d_S with $S \in \mathfrak{S}_{\tilde{\Theta}}$. Define the set of splits

$$' = \{\{\{y_2, \ldots, y_i\}, Y \backslash \{y_2, \ldots, y_i\}\} \mid 3 \leq i \leq n - 1\}.$$

By a case analysis similar to the one applied in the proof of Lemma 4.7 we obtain the following:

- $\hat{Q}[d_S](y_2, y_3) = \hat{Q}[d_S](y_1, y_3)$ if $S \in \mathfrak{S}_{\tilde{\Theta}} \backslash '$, and

Table 1: List of cases in the proof of Lemma 4.7

Case	i	j	Case	i	j
(i)	$i = 1$	$1 \leq j < r$	(iv)	$1 < i \leq r$	$r \leq j < n$
(ii)	$i = 1$	$r \leq j < n$	(v)	$r < i < n$	$i \leq j < n$
(iii)	$1 < i < r$	$i \leq j < r$			

Table 2: Precomputed expressions used in the proof of Lemma 4.7

Case	$\sum_{l=1}^{r-1} d_S(x_l, x_{l+1})$	$d_S(x_1, x_r)$	$\sum_{l=1}^{n} d_S(x_1, x_l)$
(i)	1	1	$n - j$
(ii)	0	0	$n - j$
(iii)	2	0	$j - i + 1$
(iv)	1	1	$j - i + 1$
(v)	0	0	$j - i + 1$
Case	$\sum_{l=2}^{r-1}\sum_{k=1}^{n} d_S(x_l, x_k)$		$\sum_{l=1}^{n} d_S(x_r, x_l)$
(i)	$(j - 1)(n - j) + (r - j - 1)j$		j
(ii)	$(r - 2)(n - j)$		$n - j$
(iii)	$(j - i + 1)(n - 2j + 2i + r - 4)$		$j - i + 1$
(iv)	$(i - 2)(j - i + 1) + (r - i)(i - 1 + n - j)$		$i - 1 + n - j$
(v)	$(r - 2)(j - i + 1)$		$j - i + 1$

• $\hat{Q}[d_S](\gamma_2, \gamma_3) < \hat{Q}[d_S](\gamma_1, \gamma_3)$ if $S \in$ '.

But then, since $\hat{Q}[d](\gamma_1, \gamma_3)$ is minimum, $\hat{Q}[d](\gamma_2, \gamma_3) = \hat{Q}[d](\gamma_1, \gamma_3)$. Thus, by the above strict inequality, for every split $S \in$ ' we must have $\omega(S) = 0$. Hence, ω is compatible with $\tilde{\Theta}'$.

Case 2: $C_1 = \{\gamma_1, \gamma_2\}$, $C_2 = \{\gamma_3, \gamma_4\}$, $x_1 = \gamma_1$, $x_2 = \gamma_4$ and $n \geq 5$. We want to show that $\tilde{\Theta}' = \gamma_2, \gamma_1, \gamma_4, \gamma_3, \gamma_5, \ldots, \gamma_n$ is compatible with ω. A similar argument to the one used in Case 1 shows that for every split S in

$$' = \{\{\{\gamma_2, \ldots, \gamma_i\}, Y\backslash\{\gamma_2, \ldots, \gamma_i\}\} | 3 \leq i \leq n - 1\} \cup \{\{\{\gamma_4, \ldots, \gamma_i\}, Y\backslash\{\gamma_2, \ldots, \gamma_i\}\} | 5 \leq i \leq n\}$$

we must have $\omega(S) = 0$. Thus, ω is compatible with $\tilde{\Theta}'$. ■

Now, by Proposition 4.10, we can apply the induction hypothesis and conclude that the recursive call FINDORDERING($\mathfrak{C}'$, d) will return an ordering Θ compatible with $\mathfrak{C}'$ and d. Since Θ will order C' according to $\Theta_{C'}$ (or its reverse), we have that Θ is compatible with C_1 and C_2. Thus Θ is compatible with $\mathfrak{C}$ and d, completing the proof of Theorem 4.2. □

Remark 4.11 Note that we have shown that Corollary 4.9 holds under the assumption that (in view of line 6) every cluster in $\mathfrak{C}$ contains at most two elements. However, it is possible to prove this result in the more general setting where clusters can have arbitrary size. In principle, this could yield a consistent variation of the Neighbor-Net algorithm that is analogous to the recently introduced QNet algorithm [16], where, instead of reducing the size of clusters when they have more than two elements, the reduction case is skipped entirely and clusters are pairwise combined until only one cluster is left. However, we suspect that such a method would probably not work well in practice since the reduced distances have smaller variance than the original distances.

References

1. Felsenstein J: *Inferring phylogenies* Sinauer Associates; 2003.
2. Bryant D, Moulton V: **NeighborNet: An agglomerative method for the construction of phylogenetic networks.** *Molecular Biology and Evolution* 2004, **21:**255-265.
3. Saitou N, Nei M: **The neighbor-joining method: A new method for reconstructing phylogenetic trees.** *Molecular Biology and Evolution* 1987, **4(4):**406-425.
4. Hu J, Fu HC, Lin CH, Su HJ, Yeh HH: **Reassortment and Concerted Evolution in Banana Bunchy Top Virus Genomes.** *Journal of Virology* 2007, **81:**1746-1761.
5. Lacher D, Steinsland H, Blank T, Donnenberg M, Whittam T: **Sequence Typing and Virulence Gene Allelic Profiling.** *Journal of Bacteriology* 2007, **189:**342-350.
6. Kilian B, Ozkan H, Deusch O, Effgen S, Brandolini A, Kohl J, Martin W, Salamini F: **Independent Wheat B and G Genome Origins in Outcrossing Aegilops Progenitor Haplotypes.** *Molecular Biology Evolution* 2007, **24:**217-227.
7. Hamed MB: **Neighbour-nets portray the Chinese dialect continuum and the linguistic legacy of China's demic history.** *Proc Royal Society B: Biological Sciences* 2005, **272:**1015-1022.
8. Dress A, Huson D, Moulton V: **Analyzing and visualizing sequence and distance data using SplitsTree.** *Discrete Applied Mathematics* 1996, **71:**95-110.
9. Huson D, Bryant D: **Application of Phylogenetic Networks in Evolutionary Studies.** *Molecular Biology and Evolution* 2006, **23:**254-267.
10. Bandelt HJ, Dress A: **A canonical split decomposition theory for metrics on a finite set.** *Advances in Mathematics* 1992, **92:**47-105.
11. Semple C, Steel M: *Phylogenetics* Oxford University Press; 2003.
12. Chepoi V, Fichet B: **A note on circular decomposable metrics.** *Geometriae Dedicata* 1998, **69:**237-240.
13. Christopher G, Farach M, Trick M: **The structure of circular decomposable metrics.** *Proc of European Symposium on Algorithms (ESA), Volume 1136 of LNCS, Springer* 1996:486-500.

14. Dress A, Huson D: **Constructing split graphs.** *IEEE Transactions on Computational Biology and Bioinformatics* 2004, **1(3):**109-115.
15. Kalmanson K: **Edgeconvex circuits and the travelling salesman problem.** *Canadian Journal of Mathematics* 1975, **27:**1000-1010.
16. Grünewald S, Forslund K, Dress A, Moulton V: **QNet: An agglomerative method for the construction of phylogenetic networks from weighted quartets.** *Molecular Biology and Evolution* 2007, **24:**532-538.
17. Kotetishvili M, Stine O, Kreger A, Morris J, Sulakvelidze A: **Multilocus sequence typing for characterization of clinical and environmental salmonella strains.** *Journal of Clinical Microbiology* 2002, **40:**1626-1635.