Journal of Bioinformatics and Computational Biology
Vol. 12, No. 5 (2014) 1450024 (27 pages)

DOI: 10.1142/S0219720014500243

Imperial College Press
www.icpress.co.uk

A new efficient algorithm for inferring explicit hybridization networks following the Neighbor-Joining principle

Matthieu Willems*, Nadia Tahiri† and Vladimir Makarenkov‡

Département d'informatique
Université du Québec à Montréal
Case postale 8888, Succursale Centre-ville
Montréal (Québec) H3C 3P8 Canada
**matthieu.willems@polytechnique.org*
†*tahiri.nadia@courrier.uqam.ca*
‡*makarenkov.vladimir@uqam.ca*

Received 26 May 2014
Revised 13 August 2014
Accepted 13 August 2014
Published 15 September 2014

Several algorithms and software have been developed for inferring phylogenetic trees. However, there exist some biological phenomena such as hybridization, recombination, or horizontal gene transfer which cannot be represented by a tree topology. We need to use phylogenetic networks to adequately represent these important evolutionary mechanisms. In this article, we present a new efficient heuristic algorithm for inferring hybridization networks from evolutionary distance matrices between species. The famous Neighbor-Joining concept and the least-squares criterion are used for building networks. At each step of the algorithm, before joining two given nodes, we check if a hybridization event could be related to one of them or to both of them. The proposed algorithm finds the exact tree solution when the considered distance matrix is a tree metric (i.e. it is representable by a unique phylogenetic tree). It also provides very good hybrids recovery rates for large trees (with 32 and 64 leaves in our simulations) for both distance and sequence types of data. The results yielded by the new algorithm for real and simulated datasets are illustrated and discussed in detail.

Keywords: Hybridization networks; Neighbor-Joining; phylogenetic trees.

1. Introduction

The evolution of species is commonly modeled by means of phylogenetic (i.e. additive) trees, in which leaves represent contemporary species, internal nodes — their ancestors, branches — the ancestor–descendant relationships between species and branch lengths — the evolutionary time. There exist many algorithms for inferring phylogenetic trees from distance or sequence datasets. Distance-based methods take

‡Corresponding author.

as input a distance matrix between species, whereas sequence-based methods take as input some DNA or protein sequences. Distance-based methods are usually much faster than the sequence-based methods. The most popular of these methods are Neighbor-Joining (NJ)[1] and Unweighted Pair Group Method with Arithmetic mean (UPGMA).[2] If we have to analyze datasets with relatively small number of species, we can also use sequence-based approaches such as maximum parsimony[3] and maximum likelihood.[4] More recently, several Bayesian methods[5] have been developed to extend maximum-likelihood estimations to a larger number of species.

Some well-known reticulate evolution phenomena, such as horizontal (lateral) gene transfer, hybridization, homoplasy, and genetic recombination cannot be modeled by a tree structure.[6–8] However, reticulate evolution has long been neglected in phylogenetic analyses. The first methods for studying the mechanisms of reticulate evolution started to appear in the mid-1970s.[6,9] Several tentative methods have been proposed for the identification of reticulate evolution in nucleotide sequences. They include displays of compatibility,[6] tests for clustering,[10] a randomization approach,[11] and an extension of the parsimony method of phylogenetic reconstruction that allows recombination.[12] Rieseberg and Morefield[13] developed a computer program, RETICLAD, allowing one to identify hybrids based on the expectation that they would combine the characters of their parents. However, the latter program could only find reticulation events between terminal branches of a tree. The popular method of split decomposition enables the representation of data in the form of a split graph revealing conflicting signals contained in the data.[14,15] A split graph is an implicit phylogenetic network, that is used to show conflicting placements of taxa. Bryant and Moulton[16,17] introduced a network-inferring method, NeighborNet, allowing the reconstruction of planar phylogenetic split networks. The latter method usually provides several decompositions of the set of species but it is very difficult to deduce explicit reticulation events (e.g. horizontal gene transfers with their directions) from theses decompositions. Gambette and Huson[18] have then improved the visualization of these decompositions. Huson and Bryant have developed the SplitsTree[17] software, which has become the most commonly used tool for inferring implicit phylogenetic networks. Legendre and Makarenkov[19–21] proposed to use reticulograms for detecting reticulation events in evolutionary data. They developed a distance-based method to infer reticulate phylogenies. The latter method uses the topology of a phylogenetic tree as backbone structure for building a reticulogram by adding reticulation branches to this tree according to an optimization criterion. Hallett and Lagergren[22] showed how horizontal gene transfer events can be detected by evaluating topological differences between species and gene trees. Makarenkov *et al.*,[23] Boc *et al.*,[24] and Boc and Makarenkov[25] have proposed algorithms for identifying and validating statistically horizontal gene transfer events from species and gene trees defined on the same set of species. These methods have been implemented in the software T-REX.[26,27] In the case of a set of individuals of the same population, Bandelt *et al.*[28] have used a parsimony criterion to construct a network from several minimal cover trees. Doyon

et al.[29] have also proposed a parsimony method to reconcile a species tree and several gene trees, by taking into account horizontal transfers, gene losses, and duplications. Huson and Rupp[30] and van Iersel *et al.*[31] have used the notion of cluster networks to reconcile several contradictory phylogenetic trees. In the case of two contradictory trees, Albrecht *et al.*[32] have proposed a partially parallel algorithm to find a minimum hybridization network from two input trees, but this algorithm remains very slow even when it is executed on a computer with multiple cores.[33] Wu[34] and Chen and Wang[35] have presented algorithms for constructing hybridization networks from more than two input trees. Van Iersel and Kelk[36] have developed a polynomial-time algorithm for inferring a phylogenetic network from a dense set of triplets. Some quartet-based methods have been also introduced for tree reconstruction by Strimmer and van Harseler,[37] and then implemented by Schmidt *et al.*[38] in the program TREE-PUZZLE. These methods have been extended to network inference in the software QNet[39] and Quartet-Net.[40] Huson and Klöpper[41,42] have designed two methods to detect recombination events from binary sequences by using general reticulate networks and galled trees (i.e. reticulate networks in which all reticulations are independent from each other). Many of the discussed techniques were tested by Woolley *et al.*[43] Mention that all of them are plausible only under specific evolutionary constraints and assumptions. It is worth noting that all of the existing methods for building explicit hybridization networks take as input either a set of contradictory trees, or a set of clusters, or a set of triplets (see Semple[44] for a more detailed description of all these structures). The main novelty of our work is that our algorithm takes as input a single distance matrix to infer from it an explicit hybridization network using the well-known principle of minimum evolution.[1]

The main goal of our article is to present a new efficient algorithm for inferring phylogenetic networks from a distance matrix between species. Our method can be seen as a generalization of the very popular NJ algorithm[1] to hybridization networks. The paper is organized as follows: In Sec. 2, we describe the phenomenon of hybridization; in Sec. 3, we recall the main features of the NJ algorithm; in Sec. 4, we define a hybridization network and prove some of its important general properties; in Sec. 5, we present the necessary least-squares formulas and discuss the network building strategy. Section 6 is dedicated to our algorithm and in Sec. 7, we provide some experimental results obtained on additive, nonadditive, and real datasets.

2. Hybridization

Hybridization is a very common mechanism of reticulate evolution. In Fig. 1, two lineages (Root-Species E2 and Root-Species E3) recombine to create a new species (Species E4). If the new species have the same number of chromosomes as the parent species, the process is called diploid hybridization. When it has the sum of the number of its parents' chromosomes, it is called polyploid hybridization. The three

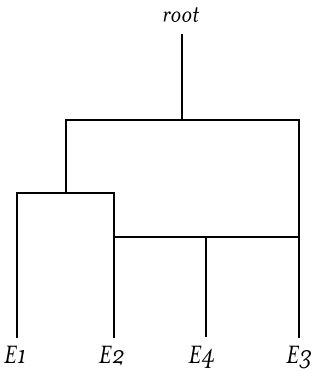


Fig. 1. An example of hybridization.

main mechanisms of hybridization are the following:

(1) Autopolyploidization is a speciation event involving the doubling of the chromosomes within a single species (intraspecific hybridization). It produces a bifurcating speciation event in a phylogenetic tree.
(2) Allopolyploidization is a type of hybridization between two species, when an offspring acquires the complete diploid chromosome complements of the two parents. In this case, the parents do not need to have the same number of chromosomes. Allopolyploidization results in instantaneous speciation because any backcrossing to the diploid parents is likely to produce a sterile triploid offspring.
(3) Diploid hybrid speciation is a normal sexual event taking place between parents from different but related species. In nearly all cases, the two parents need to have the same number of chromosomes. In this case, successful backcrossing to the parents is possible, so the hybrids have to be isolated from the parents to become new species.

Consider the problem of modeling reticulate evolution after diploid hybrid speciation. In normal diploid organisms, each chromosome consists of a pair of homologs. In the process of diploid hybridization, the hybrid inherits one of the two homologs for each chromosome from each of its two parents. Since the genes from both parents contributed to the hybrid, the evolution of genes inherited from each parent can be represented on separate trees inside a network model. Classical phylogenetic analysis of the four species involved in a hybrid speciation event (Fig. 1) will produce one of the two trees in Fig. 2.

Hybridization is very common in plants. There exist more than 70,000 natural hybrid plants,[45] and some hybrid plants can be created by humans to introduce some specific characteristics into cultivated species.[46] Hybridization is also very common among fish, amphibians, and reptiles,[47] and is rare in other groups, particularly in birds, mammals, and most arthropods. The latter groups are only occasionally

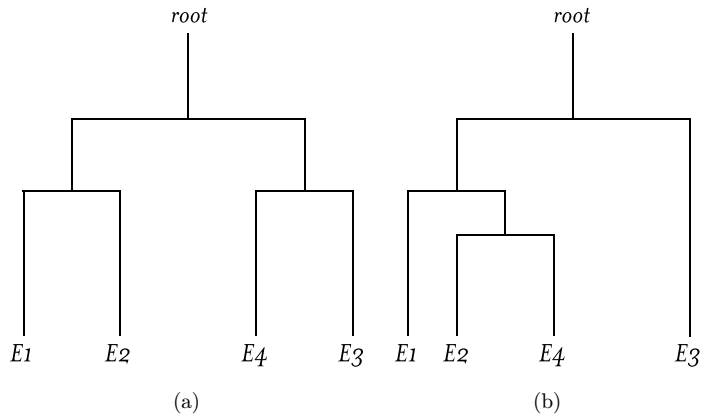


Fig. 2. Two different trees for representing the same hybridization phenomenon of Fig. 1.

affected by hybrid speciation. They usually produce triploids which can only reproduce by asexual modes.

The main goal of our article is to model and infer phylogenetic networks taking into account possible hybridization events.

3. NJ for Trees

This section starts with some basic definitions concerning phylogenetic trees.[48] The distance $d(x, y)$ between two vertices x and y in a phylogenetic tree T is defined as the sum of all branch lengths of the unique path connecting x and y in T.

Definition 1. Let X be a set of n species. A dissimilarity d on X is a nonnegative function on $X \times X$ such that for all x, y in X:

(1) $d(x, y) = d(y, x)$, and
(2) $d(x, y) = d(y, x) \geq d(x, x) = 0$.

Definition 2. A dissimilarity d on X is said to satisfy the four-point condition[49] if for all $x, y, z,$ and w in X: $d(x, y) + d(z, w) \leq \text{Max}\{d(x, z) + d(y, w); d(x, w) + d(y, z)\}$.

Definition 3. For any finite set X, an X-tree is an ordered pair (T, ϕ) consisting of a tree T, with a set of vertices V and a relation $\phi : X \to V$, such that, for all $v \in V$ with a degree at most equal to 2, $v \in \phi(X)$. An X-tree is a phylogenetic tree if ϕ is a bijection from X to the set of all leaves of T. It is said to be binary if each internal vertex has a degree equal to 3.

The main theorem relating the four-point condition and phylogenetic trees is as follows:

Theorem 1. (*Zarestskii, Buneman, Patrinos, and Hakimi, Dobson*) *Any dissimilarity satisfying the four-point condition can be represented as a phylogenetic tree*

such that for all x, y in X, $d(x, y)$ is equal to the length of the path connecting leaves x and y in T.

This dissimilarity is called an additive distance, a tree distance or a tree metric. This tree is unique.

The NJ[1] algorithm is the most popular distance-based method for inferring phylogenetic trees. Atteson[50] proved that this algorithm finds the correct phylogeny if the input distances between species are sufficiently close to the real evolutionary distances.

Throughout this article, we take as input a distance matrix $D = \{D[i][j]\}_{1\leq i\leq n;1\leq j\leq n}$ on a set of n species, and we obtain as output a network corresponding to the evolutionary history of these species. Obviously, $D[i][i] = 0$ for all $1 \leq i \leq n$, and $D[i][j] = D[j][i]$ for all $1 \leq i \leq n$ and $1 \leq j \leq n$.

NJ is a clustering algorithm which starts with a bush composed of n leaves and n branches, where n is the number of current species. This tree is gradually transformed into an unrooted phylogenetic tree with the same n leaves and with $2n - 3$ branches. The ith step consists in choosing two neighbors among $n - i + 1$ candidates. We consider all the $\frac{(n-i+1)(n-i)}{2}$ configurations similar to the one represented in Fig. 3. For each of these configurations, we calculate the branch lengths which minimize a least-squares criterion, in which we compare the input dissimilarities with the tree metric distances.

Saitou and Nei[1] showed that the sum of the branch lengths of the tree topology in Fig. 3 is equal to:

$$S_{i;j} = \frac{1}{2}D[i][j] + \frac{\sum_{1\leq k\leq n;k\neq i,j}[D[i][k] + D[j][k]]}{2(n-2)} + \frac{\sum_{1\leq k<l\leq n;k,l\neq i,j} D[k][l]}{n-2}. \quad (1)$$

We connect nodes i and j that minimize the total evolution, i.e. the sum of branch lengths $S_{i;j}$. We replace the selected nodes i and j by node X (their direct common ancestor) and obtain a distance matrix of size $n - 1$. We compute the new distances from X to the remaining leaves of the tree by using the following formula:

$$d(X, k) = \frac{1}{2}(D[i][k] + D[j][k]), \quad k \neq i, j. \quad (2)$$

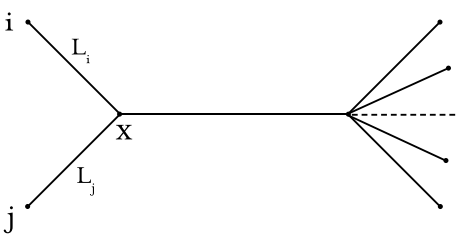


Fig. 3. Configuration where nodes i and j are chosen as neighbors.

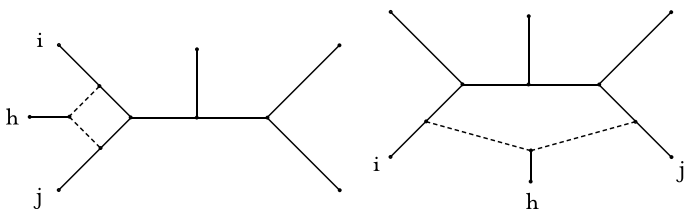


Fig. 4. Hybrids between terminal branches.

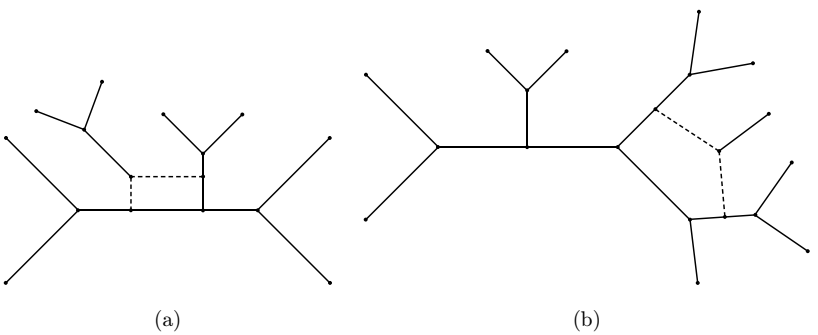


Fig. 5. Hybrids between ancestral branches.

After $n-3$ steps, we obtain an unrooted phylogenetic tree whose branch lengths are calculated at each step by using the following equations:

$$L_i = \frac{1}{2}D[i][j] + \frac{1}{2(n-2)}(P-Q), \quad L_j = \frac{1}{2}D[i][j] - \frac{1}{2(n-2)}(P-Q), \qquad (3)$$

where

$$P = \sum_{1\le k\le n, k\neq i,j} D[i][k], \quad \text{and} \quad Q = \sum_{1\le k\le n, k\neq i,j} D[j][k].$$

We adapt this algorithm to the case of hybridization networks. Note that in our model hybridization events may occur between terminal branches such as shown in Fig. 4 as well as between ancestral branches such as shown in Fig. 5. In both cases, the two parent branches may (or may not) have a direct common ancestor (see the difference between networks (a) and (b) in both figures). In all figures, hybridization (i.e. reticulation) branches are depicted by dashed lines.

4. Some Properties of Hybridization Networks

In this section, we consider hybrids between terminal branches. We describe how distances between species are defined in such a network.

4.1. *Hybrids between neighbor parents*

We take the network (a) in Fig. 4 as an example where species h is the hybrid of species i and j. We denote by X the common ancestor of species i and j (see Fig. 6). Obviously, if we remove species h, we obtain a traditional additive tree. For hybrid species h, we define a real value α between 0 and 1, which is the proportion of the hybrid's genetic inheritance coming from species i (see Fig. 6). We also need to know lengths L_i^0, L_j^0, and L_h shown in Fig. 6. The dashed reticulation branches labeled α and $1-\alpha$ have branch lengths equal to 0. The distances between hybrid h and other species in the network are defined as follows:

$$\begin{aligned} D[i][h] &= L_h + \alpha(L_i - L_i^0) + (1-\alpha)(L_j^0 + L_i), \quad \text{and then} \\ D[i][h] &= L_h + L_i - \alpha L_i^0 + (1-\alpha)L_j^0, \end{aligned} \tag{4}$$

$$\begin{aligned} D[j][h] &= L_h + (1-\alpha)(L_j - L_j^0) + \alpha(L_i^0 + L_j), \quad \text{and then} \\ D[j][h] &= L_h + L_j + \alpha L_i^0 - (1-\alpha)L_j^0, \end{aligned} \tag{5}$$

$$D[k][h] = L_h + \alpha L_i^0 + (1-\alpha)L_j^0 + d(X,k), \tag{6}$$

for any species k different from i, j, and h, where the distances $d(X,k)$ between nodes X and k are computed as in a traditional additive tree.

Since the only terms containing the hybridization parameter α, L_i^0 or L_j^0 are αL_i^0 and $(1-\alpha)L_j^0$, α is not uniquely defined. We can increase α and L_j^0 and decrease L_i^0 in such a way that we obtain exactly the same distances between species. It is no longer the case for hybrids between non-neighbor parents.

4.2. *Hybrids between non-neighbor parents*

We take the network (b) in Fig. 4 as an example where h is the hybrid of species i and j. We denote by X (respectively Y) the closest ancestor of species i (respectively j). We use the notation indicated in Fig. 7. Thus, the distances between hybrid h and

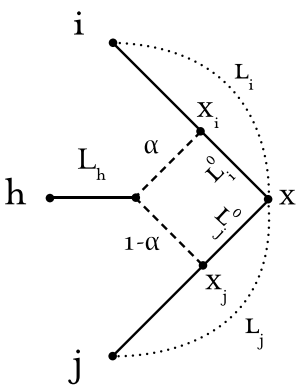


Fig. 6. Network configuration in which species h is a hybrid of two neighbor species i and j.

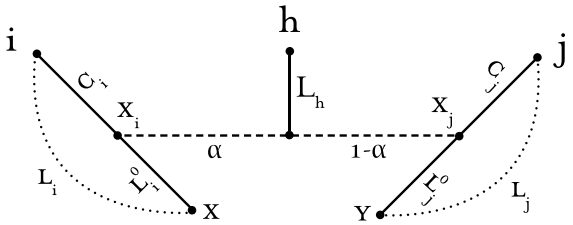


Fig. 7. Network configuration in which species h is a hybrid of two non-neighbor species i and j.

other species in the network are defined as follows:

$$\begin{aligned} D[i][h] &= L_h + \alpha(L_i - L_i^0) + (1-\alpha)(L_j^0 + d(Y,X) + L_i), \quad \text{and then} \\ D[i][h] &= L_h + L_i - \alpha L_i^0 + (1-\alpha)(L_j^0 + d(Y,X)), \end{aligned} \tag{7}$$

$$\begin{aligned} D[j][h] &= L_h + (1-\alpha)(L_j - L_j^0) + \alpha(L_i^0 + d(Y,X) + L_j), \quad \text{and then} \\ D[j][h] &= L_h + L_j + \alpha(L_i^0 + d(Y,X)) - (1-\alpha)L_j^0, \end{aligned} \tag{8}$$

$$D[k][h] = L_h + \alpha(L_i^0 + d(X,k)) + (1-\alpha)(L_j^0 + d(Y,k)) \tag{9}$$

for all species k different from i, j, and h, where the distances $d(X,k)$, $d(Y,X)$, and $d(Y,k)$ are computed as in a traditional additive tree. If we set $Y = X$ in these equations, we obtain the equations for a hybrid between neighbors. As we will show later, if species i and j are not neighbors, α is uniquely defined and can be calculated directly from the distance matrix.

4.3. *Two important properties*

In this section, we will describe two very important properties of hybridization networks that will be used in our algorithm.

Proposition 1. *If species h is the hybrid of species i and j, then for all species k different from species i, j, and h:*

$$\begin{aligned} D[i][j] + D[k][h] - D[i][h] - D[k][j] > 0, \\ D[i][j] + D[k][h] - D[j][h] - D[k][i] > 0. \end{aligned} \tag{10}$$

Moreover, if species i and j are neighbors, then for all species k different from species i, j, and h:

$$\begin{aligned} D[i][j] + D[k][h] - D[i][h] - D[k][j] = 2\alpha L_i^0, \\ D[i][j] + D[k][h] - D[j][h] - D[k][i] = 2(1-\alpha)L_j^0. \end{aligned} \tag{11}$$

Proof. Let k be a species different from species i, j, and h. Using Eqs. (7) and (9), we obtain:

$$\begin{aligned} &D[i][j] + D[k][h] - D[i][h] - D[k][j] \\ &\quad = D[i][j] + L_h + \alpha(L_i^0 + d(X,k)) + (1-\alpha)(L_j^0 + d(Y,k)) \\ &\qquad - (L_h + L_i - \alpha L_i^0 + (1-\alpha)(L_j^0 + d(Y,X))) - D[k][j]. \end{aligned}$$

If we replace $D[i][j]$ by $L_i + d(Y,X) + L_j$ and $D[k][j]$ by $d(Y,k) + L_j$, we obtain:

$$\begin{aligned} &D[i][j] + D[k][h] - D[i][h] - D[k][j] \\ &\quad = L_i + d(Y,X) + L_j + L_h + \alpha(L_i^0 + d(X,k)) \\ &\qquad + (1-\alpha)(L_j^0 + d(Y,k)) - (L_h + L_i - \alpha L_i^0 + (1-\alpha)(L_j^0 + d(Y,X))) \\ &\qquad - (d(Y,k) + L_j) = \alpha(d(Y,X) + d(X,k) - d(Y,k) + 2L_i^0) > 0. \end{aligned}$$

Indeed, $d(Y,X) + d(X,k) - d(Y,k)$ is nonnegative according to the triangle inequality.

In the same way, we have:

$$\begin{aligned} &D[i][j] + D[k][h] - D[j][h] - D[k][i] \\ &\quad = (1-\alpha)(d(Y,X) + d(Y,k) - d(X,k) + 2L_j^0) > 0. \end{aligned}$$

Moreover, if species i and j are neighbors, $X = Y$, then we find Eq. (11) by replacing X by Y in the formulas above. □

Definition 4. For all triplets of species i, j, and h, we define $\mathrm{MIN}_{i,j}^h$ as the minimum of all values $D[i][j] + D[k][h] - D[i][h] - D[k][j]$ and $D[i][j] + D[k][h] - D[j][h] - D[k][i]$, for all species k different from species i, j, and h.

It is worth noting that in an additive tree (without a hybrid), for all triplets of species i, j, h, at least one of the values $D[i][j] + D[k][h] - D[i][h] - D[k][j]$ or $D[i][j] + D[k][h] - D[j][h] - D[k][i]$ is nonpositive according to the four-point condition. Then all the values $\mathrm{MIN}_{i,j}^h$ are nonpositive.

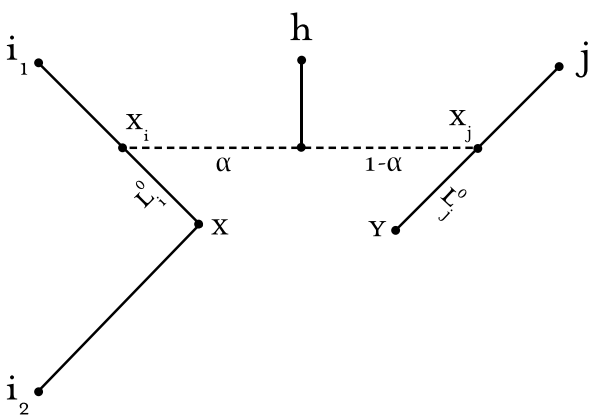


Fig. 8. Hybrid h whose parent i_1 has a direct neighbor i_2.

In the same way, we can prove the following proposition:

Proposition 2. *If species h is the hybrid of species i_1 and j, and if $i_2 \neq j$ is the direct neighbor of i_1 (see Fig. 8), then for all species k different from i_1, i_2, h, and j:*

$$\begin{aligned} D[i_2][h] + D[i_1][k] - D[k][h] - D[i_1][i_2] > 0, \\ D[i_2][h] + D[i_1][k] - D[i_1][h] - D[k][i_2] > 0 \end{aligned} \tag{12}$$

and also $\mathrm{MIN}_{i_2,h}^{i_1} > 0$.

5. Identification of Hybrids

If species h is the hybrid of species i and j in an additive network, we have the following system of equations, where we use the length C_i (respectively C_j) between species i and X_i (respectively j and X_j), as shown in Fig. 7:

$$D[i][h] = L_h + \alpha C_i - (1-\alpha)C_j + (1-\alpha)D[i][j], \tag{13}$$

$$D[j][h] = L_h - \alpha C_i + (1-\alpha)C_j + \alpha D[i][j], \tag{14}$$

$$D[k][h] = L_h - \alpha C_i - (1-\alpha)C_j + \alpha D[i][k] + (1-\alpha)D[j][k] \tag{15}$$

for all species k different from i, j, and h.

In a general network, we computed the least-squares solutions to this system of equations, and we found the following formulas:

$$L_h = \frac{1}{2}(D[i][h] + D[j][h] - D[i][j]), \tag{16}$$

$$\begin{aligned} \alpha C_i = &-\frac{1}{2(n-3)}\left(\sum_{k' \neq i,j,h}(D[k'][h] - \alpha D[k'][i] - (1-\alpha)D[k'][j])\right) \\ &+ \frac{1}{2}(D[i][h] - (1-\alpha)D[i][j]), \end{aligned} \tag{17}$$

$$\begin{aligned} (1-\alpha)C_j = &-\frac{1}{2(n-3)}\left(\sum_{k' \neq i,j,h}(D[k'][h] - \alpha D[k'][i] - (1-\alpha)D[k'][j])\right) \\ &+ \frac{1}{2}(D[j][h] - \alpha D[i][j]). \end{aligned} \tag{18}$$

If we set $A = -\frac{1}{2(n-3)}(\sum_{k' \neq i,j,h}(D[k'][h] - \alpha D[k'][i] - (1-\alpha)D[k'][j]))$, we obtain:

$$\alpha C_i = A + \frac{1}{2}(D[i][h] - (1-\alpha)D[i][j]),$$

$$(1-\alpha)C_j = A + \frac{1}{2}(D[j][h] - \alpha D[i][j]).$$

If we replace L_h, αC_i, and $(1-\alpha)C_j$ by these formulas in Eqs. (13)–(15), we obtain:

$$\begin{aligned}
D[i][h] &= \frac{1}{2}(D[i][h] + D[j][h] - D[i][j]) \\
&\quad + \frac{1}{2}(D[i][h] - (1-\alpha)D[i][j] - D[j][h] + \alpha D[i][j]) + (1-\alpha)D[i][j], \\
D[j][h] &= \frac{1}{2}(D[i][h] + D[j][h] - D[i][j]) \\
&\quad - \frac{1}{2}(D[i][h] - (1-\alpha)D[i][j] - D[j][h] + \alpha D[i][j]) + \alpha D[i][j], \\
D[k][h] &= \frac{1}{2}(D[i][h] + D[j][h] - D[i][j]) - 2A \\
&\quad - \frac{1}{2}(D[i][h] - (1-\alpha)D[i][j] + D[j][h] \\
&\quad - \alpha D[i][j]) + \alpha D[i][k] + (1-\alpha)D[j][k].
\end{aligned}$$

After simplification, we obtain twice $0 = 0$, and

$$\begin{aligned}
D[k][h] &= -2A + \alpha D[i][k] + (1-\alpha)D[j][k]. \\
D[k][h] &= \frac{1}{n-3}\left(\sum_{k' \neq i,j,h} (D[k'][h] - \alpha D[k'][i] - (1-\alpha)D[k'][j])\right) \\
&\quad + \alpha D[i][k] + (1-\alpha)D[j][k]. \\
D[k][h] &= \frac{1}{n-3}\left(\sum_{k' \neq i,j,h} (D[k'][h] - D[k'][j])\right) \\
&\quad + \frac{\alpha}{n-3}\left(\sum_{k' \neq i,j,h} (D[k'][j] - D[k'][i])\right) \\
&\quad + \alpha D[i][k] + (1-\alpha)D[j][k].
\end{aligned}$$

If we set

$$S_h = \frac{\sum_{k' \neq i,j,h} D[k'][h]}{n-3}, \quad S_i = \frac{\sum_{k' \neq i,j,h} D[k'][i]}{n-3}, \quad S_j = \frac{\sum_{k' \neq i,j,h} D[k'][j]}{n-3}, \quad (19)$$

we obtain:

$$\begin{gathered}
D[k][h] = S_h - S_j + \alpha(S_j - S_i) + \alpha D[i][k] + (1-\alpha)D[j][k], \quad \text{and then} \\
D[k][h] - D[j][k] = S_h - S_j + \alpha(S_j - S_i + D[i][k] - D[j][k]).
\end{gathered}$$

Now, we have to find an optimal value of α allowing us to minimize the following function:

$$\sum_{k \neq i,j,h} (Y_k - S_h + S_j - \alpha X_k)^2,$$

where

$$Y_k = D[k][h] - D[j][k], \quad X_k = S_j - S_i + D[i][k] - D[j][k]. \tag{20}$$

If we differentiate according to α, we obtain:

$$\sum_{k \neq i,j,h} X_k(Y_k - S_h + S_j - \alpha X_k) = 0$$

and then

$$\alpha^* = \frac{\sum_{k \neq i,j,h} X_k(Y_k - S_h + S_j)}{\sum_{k \neq i,j,h} X_k X_k}.$$

Definition 5. For all triplets of species i, j, h, we define the degree of hybridation, $\alpha^h_{i,j}$, of h as a hybrid of i and j, by the following formula:

$$\alpha^h_{i,j} = \frac{\sum_{k \neq i,j,h} X_k(Y_k - S_h + S_j)}{\sum_{k \neq i,j,h} X_k X_k},$$

where S_h, S_j, X_k, and Y_k are defined by Eqs. (19) and (20).

If $\sum_{k \neq i,j,h} X_k X_k = 0$ (which is the case when h is the hybrid of neighbor species i and j in an additive network), the optimal value of α cannot be determined and we set $\alpha^h_{i,j} = 0.5$.

We also define $L^h_{i,j}$ by the following formula:

$$L^h_{i,j} = \frac{\sum_{k \neq i,j,h} (Y_k - S_h + S_j - \alpha^h_{i,j} X_k)^2}{(n-3)}.$$

Remark 1. The closer $L^h_{i,j}$ is to 0, the more likely species h is the hybrid of species i and j.

Our strategy to identify hybrids is the following. First, we determine the couple (i_1, i_2), which minimizes $S_{i;j}$ according to the NJ classical criterion.

Then, we identify the species h that is the most likely to be a hybrid between i_1 or i_2 and any other species j. We notice that in an additive tree, if i_1 and i_2 are true neighbors, we have the following equations:

$$D[i_1][i_2] = L_{i_1} + L_{i_2},$$
$$D[i_1][k] - D[i_2][k] = L_{i_1} - L_{i_2},$$

for all species k different from i_1 and i_2. Then, we obtain the following equations:

$$D[i_2][k] + D[i_1][k'] - D[i_2][k'] - D[i_1][k] = 0,$$

for all species k and k' different from i_1 and i_2. However, if species h is the hybrid of species i_1 (respectively i_2) and any species j, according to Proposition 2, we have:

$$D[i_2][h] + D[i_1][k'] - D[i_2][k'] - D[i_1][h] > 0,$$
$$(D[i_2][h] + D[i_1][k'] - D[i_2][k'] - D[i_1][h] < 0, \text{respectively}),$$

for all species k' different from i_1, i_2, and h. Then, we define $\Sigma^h_{i_1,i_2}$ as follows:

$$\Sigma^h_{i_1,i_2} = \frac{\sum_{k \neq i_1,i_2} (D[i_2][h] + D[i_1][k] - D[i_2][k] - D[i_1][h])}{n-2}$$

and we choose the species h_H that maximizes the absolute value of $\Sigma^h_{i_1,i_2}$.

If $\Sigma^{h_H}_{i_1,i_2} > 0$ ($\Sigma^{h_H}_{i_1,i_2} < 0$, respectively), we consider i_1 (respectively i_2) as one of the two parents of h_H. We set $i_H = i_1$ (respectively $i_H = i_2$).

Then, we need to determine the second parent of h_H. We compute the smallest value $L^{h_H}_{i_H,j_H}$, of all possible values $L^{h_H}_{i_H,j}$ for all species j such that: $\text{MIN}^{h_H}_{i_H,j} > 0$ and $\alpha_{\min} < \alpha^{h_H}_{i_H,j} < \alpha_{\max}$, where the thresholds $\alpha_{\min}$ and $\alpha_{\max}$ ($0 < \alpha_{\min} < \alpha_{\max} < 1$) are some parameters which can be suggested by the user.

If $L^{h_H}_{i_H,j_H} < (\Sigma^{h_H}_{i_1,i_2})^2$, then h_H is identified as the hybrid of species i_H and j_H. The detailed scheme of our algorithm is given below.

6. Algorithm for Inferring Hybridization Networks

In this section, we introduce a new algorithm for inferring hybridization networks based on the NJ principle. This algorithm takes as input a distance matrix $D = \{D[i][j]\}_{1 \leq i \leq n; 1 \leq j \leq n}$ on a set of n species and two real values $\alpha_{\min}$ and $\alpha_{\max}$ such that $0 < \alpha_{\min} < \alpha_{\max} < 1$.

ALGORITHM

- $n_A = n$
- $D_A = D$
- **While** $(n_A > 4)$

  (1) We determine the couple (i_1, i_2), that minimizes $S_{i;j}$.
  (2) We choose the species h_H that maximizes the absolute value of $\Sigma^h_{i_1,i_2}$.
  (3) **If** $\Sigma^{h_H}_{i_1,i_2} > 0$, **then** $i_H = i_1$. **Else if** $\Sigma^{h_H}_{i_1,i_2} < 0$, **then** $i_H = i_2$.
  (4) We compute the smallest value $L^{h_H}_{i_H,j_H}$ of all values $L^{h_H}_{i_H,j}$ for all species j such that: $\text{MIN}^{h_H}_{i_H,j} > 0$ and $\alpha_{\min} < \alpha^{h_H}_{i_H,j} < \alpha_{\max}$.
  (5) **If** $(L^{h_H}_{i_H,j_H} \leq (\Sigma^{h_H}_{i_1,i_2})^2)$, **then** h_H is identified as the hybrid of i_H and j_H. We remove from D_A the row and the column corresponding to h_H. We keep in memory the length

$$L^H_{h_H} = \frac{1}{2}(D[i_H][h_H] + D[j_H][h_H] - D[i_H][j_H]), \tag{21}$$

  which can be deduced from Eq. (16).
  (6) **Else**, species i_1 and i_2 are considered as neighbors. We replace rows and columns corresponding to i_1 and i_2 in D_A by a row and a column corresponding to their ancestor X. Distances from X to remaining leaves are calculated using Eq. (2). We keep in memory the lengths L_{i_1} and L_{i_2} given by Eq. (3).
  (7) $n_A \leftarrow n_A - 1$.

- **End(While)**
- At the end of this loop, we have one quadruplet of nodes remaining. We use the standard NJ algorithm to determine the tree structure involving these four species.

The output of our algorithm is either a classical phylogenetic tree with n leaves, or a hybridization network with the same n terminal nodes. Its time complexity is $O(n^3)$ as in the standard NJ algorithm.

Remark 2. We do not include the iteration $n_A = 4$ in the loop since in this case the hybridization network corresponding to a distance matrix is not unique, as it is illustrated by the example of Fig. 9, where networks (a) and (b) correspond to the same distance matrix

$$D = \begin{pmatrix} 0 & 3 & 3 & 3 \\ 3 & 0 & 4 & 3 \\ 3 & 4 & 0 & 3 \\ 3 & 3 & 3 & 0 \end{pmatrix}.$$

In these networks, $\alpha = 0.5$ and all branch lengths are equal to 1. The length of the dashed branches is equal to 0.

This algorithm has been implemented in the $C++$ programming language and a Monte Carlo simulation study was carried out to assess its performances.

7. Results of Simulations

In our simulations, we consider as a true positive any hybrid that is identified as a hybrid, even if its parents are not correctly identified. We give more precise results about the identification of hybrids' parents in some particular cases. We consider as a false positive any identified hybrid that is not a true hybrid, even if one or two of the identified parents are true hybrids. The true positive rate is computed as the number of true positives divided by the number of true hybrids. The false positive rate is computed as the number of false positives divided by the number of nonhybrid species in the tree.

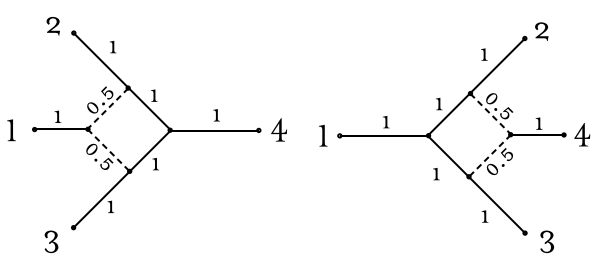


Fig. 9. Two networks corresponding to the same distance matrix of size 4.

In our simulations, we set $\alpha_{\min} = 1 - \alpha_{\max}$. Then, we defined DIFFMAX $= \alpha_{\max} - 0.5$.

7.1. *Simulations with additive networks*

7.1.1. *A theoretical result for trees*

Proposition 3. *If the input distance matrix is a tree metric, our algorithm finds the exact phylogenetic tree corresponding to this matrix.*

Proof. If the input distance matrix is a tree metric, then for all triplets of species we have $\text{MIN}_{i,j}^{h} \leq 0$. Consequently, our algorithm does not find any hybrid, and the identification of neighbors is exactly the same as in the standard NJ algorithm. Thus, the correct phylogenetic tree is recovered.[51] □

7.1.2. *Simulation with hybrids between terminal branches*

In our simulations, we used an algorithm generating random phylogenetic trees available on the T-REX website.[26,27] This algorithm takes as input the size of the tree n and the average branch length $\bar{l}$, and gives as output a random binary phylogenetic tree with n leaves constructed according to the procedure described by Kuhner and Felsenstein.[52] In this way, We generated 1,000 unrooted trees for each considered tree size $n = 8$, $n = 16$, $n = 32$, and $n = 64$, with $\bar{l} = 0.1$. Then, the following procedure was used to add a hybrid: We randomly selected two integers $1 \leq i < j \leq n$, two real values β and γ between 0 and 1, and a real value x from an exponential distribution with mean 0.1. We added a species $h = n + 1$ by using Eqs. (13)–(15) with $C_i = \beta L_i$, $C_j = \gamma L_j$, and $L_h = x$. We added from 1 to 5 hybrids to each of the trees. We carried out three series of simulations for $\alpha = 0.3$, $\alpha = 0.4$, and $\alpha = 0.5$, with DIFFMAX $= 0.25$. We computed the average true positive and false positive rates for all sizes of trees and all numbers of hybrids. These results are shown in Fig. 10.

The best results were obtained for greater values of n, for smaller numbers of hybrids, and for the values of α close to 0.5. We can also observe that the false positive rate is very close to 0, and that the true positive rates are close to 100% for $n = 64$. Most false negatives are hybrids between neighbors or between very close parents. That is the reason why the identification of hybrids is more complicated for smaller values of n. In the case of neighbor parents, the following issue can appear. If species h is the hybrid of neighbor species i and j, the NJ algorithm can identify i and h (or j and h) as neighbors. Thus, species h is identified as a parent of a hybrid and not as a hybrid.

It is worth mentioning that all true positives in this simulation were identified with both correct parents and with the correct value of α.

7.1.3. *Simulation with hybrids having two descendants*

We also carried out a simulation with a hybrid having two descendants as in the configuration (a) in Fig. 11. The obtained true positive and false positive rates were

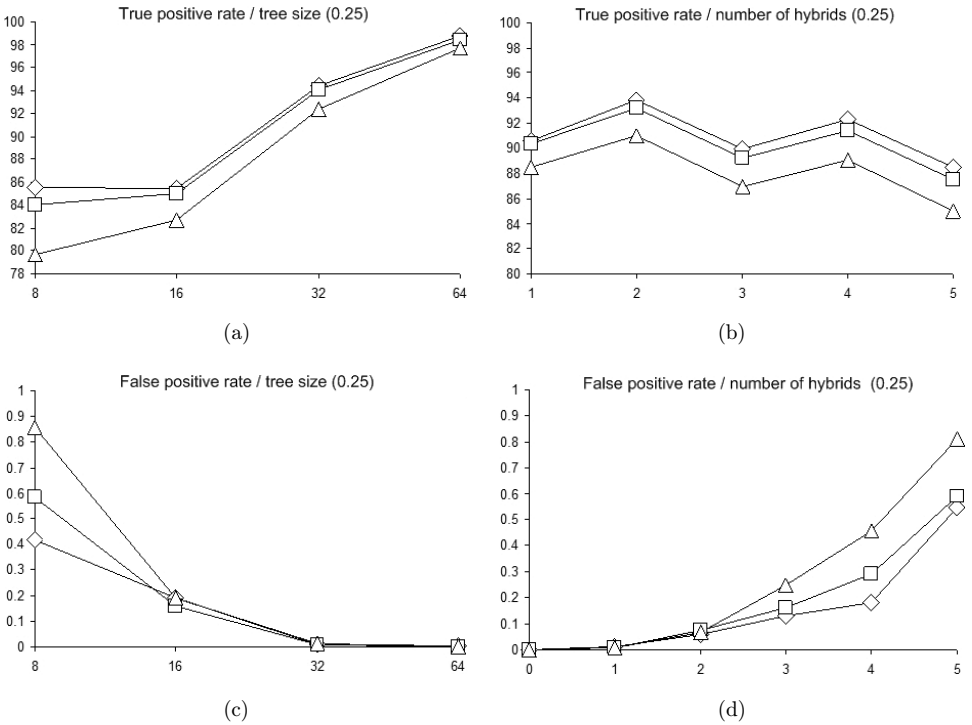


Fig. 10. Average simulation results for additive networks with hybridization level $\alpha = 0.3$ (Δ), $\alpha = 0.4$ ($\Box$), and $\alpha = 0.5$ ($\diamond$), and with DIFFMAX $= 0.25$. Figure (a) (respectively (c)) shows the true (respectively false) positive rate as a function of the tree size. Figure (b) (respectively (d)) shows the true (respectively false) positive rate as a function of the number of hybrids.

very similar to those shown in Fig. 10. However, the detection of correct parents was not systematic. Table 1 reports the identification of hybrids' parents for $\alpha = 0.5$. If the algorithm identifies a set of at least two species as a hybrid, we can solve this problem by applying the algorithm once more, replacing this set of species by their

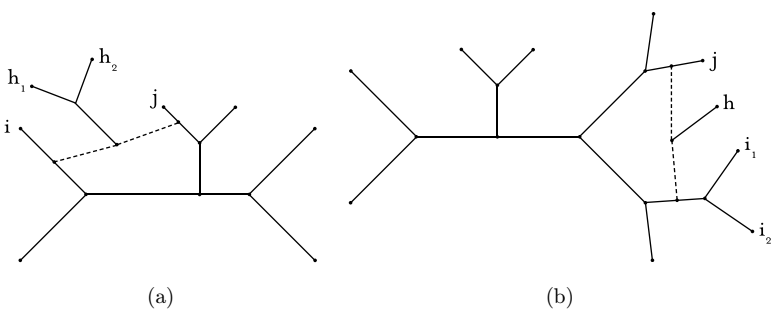


Fig. 11. Two network topologies used in our simulations with additive networks.

Table 1. Identification of hybrids' parents for additive networks with one hybrid having two descendants.

Tree size	$n = 8$	$n = 16$	$n = 32$	$n = 64$
True positives with both correct parents	75%	75%	69%	49%
True positives with only one correct parent	3%	11%	25%	49%
True positives with no correct parents	0%	0%	2%	0%

common ancestor. In this case, we will find both correct parents like in the previous simulation.

7.1.4. *Simulation with hybrids between nonterminal branches*

We also run a simulation involving a hybrid with one parent having two descendants as shown in Fig. 11(b). The true positive rates in this simulation were lower than in the two previous simulations, as it is shown in Table 2 for $n = 32$ and $\alpha = 0.5$. These results are due to the fact that the condition $\text{MIN}_{i_H,j}^{h_H} > 0$ at step 4 of our algorithm does not hold for hybrids between nonterminal branches. Then, hybrid h is detected only if species i_1 and i_2 are joined before possible detection of h. If we replace the condition $\text{MIN}_{i_H,j}^{h_H} > 0$ by $\text{MIN}_{i_H,j}^{h_H} > -0.01$, for example, we will obtain almost the same true positive rates as in the two previous simulations. However, the false positive rate will be higher.

7.2. *Simulations with nonadditive networks (i.e. with sequence-based networks)*

As previously, we generated 1,000 unrooted trees for each size $n = 8$, $n = 16$, $n = 32$, and $n = 64$, with $\bar{l} = 0.1$. Then, using Seq-Gen,[53] we simulated the evolution of nucleotide sequences of length $N = 1{,}000$ along these trees, by using the Kimura-2-parameter substitution model.[54] Thus, for each generated tree we obtained n sequences (one sequence per species) of size N. Then, the hybrids were added to the data as follows. We randomly chose two integers $1 \leq i < j \leq n$. Let α be the selected degree of hybridation. We created a new hybrid sequence with the first $\alpha \times N$ nucleotides of sequence i to which we added the last $(1 - \alpha) \times N$ nucleotides of sequence j. This sequence was added to the n original sequences. In our simulations, we considered $\alpha = 0.3$, $\alpha = 0.4$, and $\alpha = 0.5$, and we added to trees 0 to 5 hybrid species. Then, we used the Phylip package[55] to obtain a distance matrix from each

Table 2. True positive rates for $n = 32$ and one hybrid with hybridization level $\alpha = 0.5$ in additive networks.

Tree size	$n = 8$	$n = 16$	$n = 32$	$n = 64$
Hybrids between terminal branches	81%	85%	98%	99%
Hybrids with one parent having two descendants	52%	64%	66%	79%

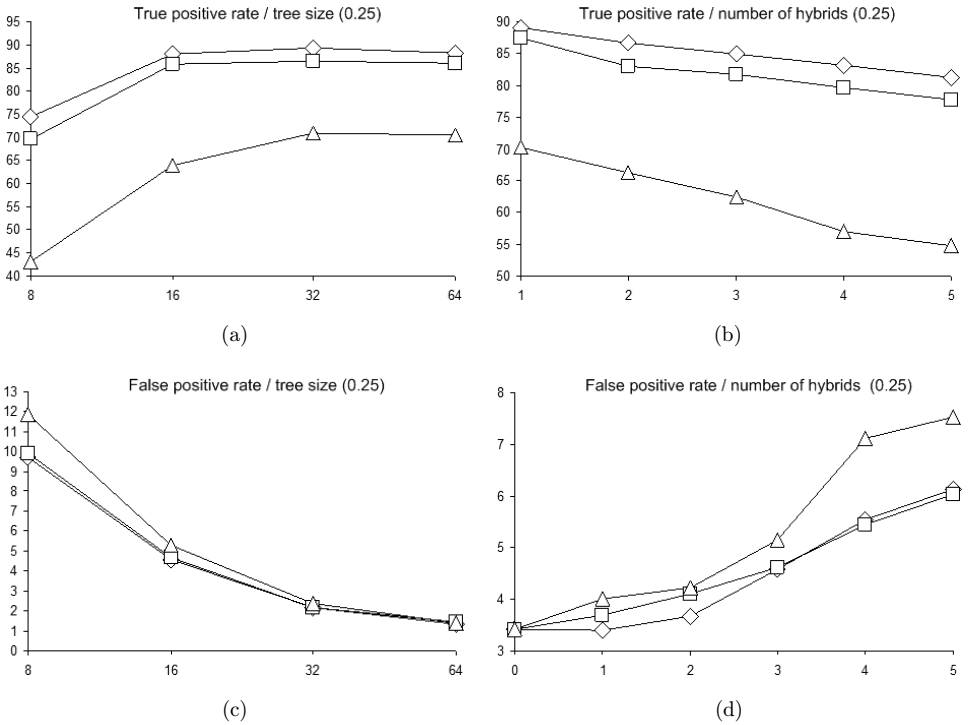


Fig. 12. Average simulation results for nonadditive networks with hybridization level $\alpha = 0.3$ (Δ), $\alpha = 0.4$ ($\Box$), and $\alpha = 0.5$ ($\diamond$), and with DIFFMAX $= 0.25$. Figure (a) (respectively (c)) shows the true (respectively false) positive rate as a function of the tree size. Figure (b) (respectively (d)) shows the true (respectively false) positive rate as a function of the number of hybrids.

set of sequences, by using the Kimura-2-parameter substitution model. Thus, for each size n, we obtained 1,000 matrices corresponding to the original trees and 15,000 matrices corresponding to networks having 1 to 5 hybrids with three different values of α. The obtained results are shown in Fig. 12 for DIFFMAX $= 0.25$ and in Fig. 13 for DIFFMAX $= 0.35$. We used two different values of DIFFMAX because there was a significant difference in the true positive rates in this simulation (this difference was much smaller in the simulations with additive data).

The greatest true positive rates and the lowest false positive rates were provided by the new algorithm for $n = 32$ and $n = 64$. The identification of hybrids was much more difficult for $\alpha = 0.3$, even though the results are much better for DIFFMAX $= 0.35$.

Table 3 reports the results concerning the identification of hybrids' parents for $\alpha = 0.5$. We can observe that the identification of both parents is more difficult for greater values of n.

Table 4 reports the number of iterations after which each hybrid was found in trees and in networks with one hybrid and with hybridization level $\alpha = 0.5$. Mention that true positives are generally detected after a much smaller number of iterations

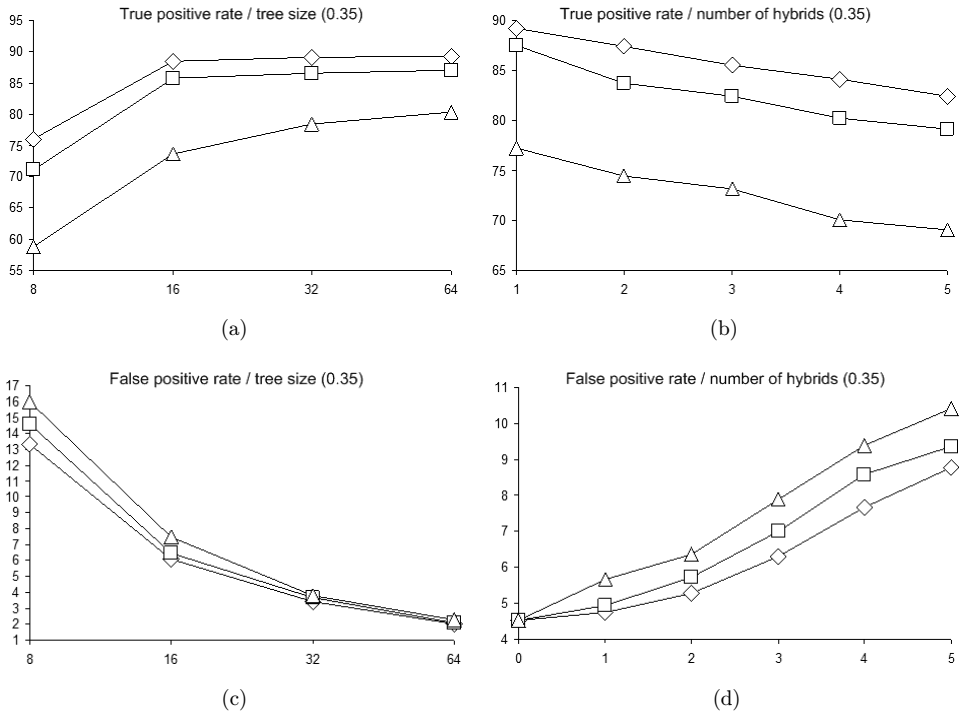


Fig. 13. Average simulation results for nonadditive networks with hybridization level $\alpha = 0.3$ (Δ), $\alpha = 0.4$ ($\square$), and $\alpha = 0.5$ ($\diamond$), and with DIFFMAX $= 0.35$. Figure (a) (respectively (c)) shows the true (respectively false) positive rate as a function of the tree size. Figure (b) (respectively (d)) shows the true (respectively false) positive rate as a function of the number of hybrids.

Table 3. Identification of hybrids' parents in nonadditive networks with one hybrid and hybridization level $\alpha = 0.5$.

Tree size	$n = 8$	$n = 16$	$n = 32$	$n = 64$
True positives with both correct parents	86%	88%	77%	53%
True positives with only one correct parent	1%	3%	11%	30%
True positives with no correct parents	0%	0%	2%	6%

Table 4. Average number of iterations μ (and the corresponding standard deviation σ) after which hybrids were detected in networks with one hybrid and hybridization level $\alpha = 0.5$.

Tree size	True positives with both parents	False positives
$n = 8$	$\mu = 1.5$, $\sigma = 1.4$	$\mu = 2.3$, $\sigma = 0.9$
$n = 16$	$\mu = 4.1$, $\sigma = 3.0$	$\mu = 8.9$, $\sigma = 2.6$
$n = 32$	$\mu = 10.0$, $\sigma = 7.6$	$\mu = 22.1$, $\sigma = 6.7$
$n = 64$	$\mu = 22.7$, $\sigma = 13.7$	$\mu = 50.7$, $\sigma = 13.7$

than false positives. Thus, the number of iterations could be an interesting criterion to consider to distinguish between true positives and false positives.

Notice that the values of α determined by our algorithm were very close to the simulated values. For example, for the case of $n = 32$ and one hybrid, we found $\mu_\alpha = 0.493$ and $\sigma_\alpha = 0.034$ for $\alpha = 0.5$, $\mu_\alpha = 0.398$ and $\sigma_\alpha = 0.034$ for $\alpha = 0.4$, $\mu_\alpha = 0.302$ and $\sigma_\alpha = 0.033$ for $\alpha = 0.3$, where μ_α (respectively, σ_α) is the mean (respectively, the standard deviation) of α found for true positive hybrids with both correct parents.

7.3. *Experiments with real data*

We tested our new algorithm on real data. We considered the dataset of restriction maps of the rDNA cistron of 12 species of mosquitoes constructed using eight recognition restriction enzymes.[56] A total of 26 sites were scored. This dataset is presented in Table 5.

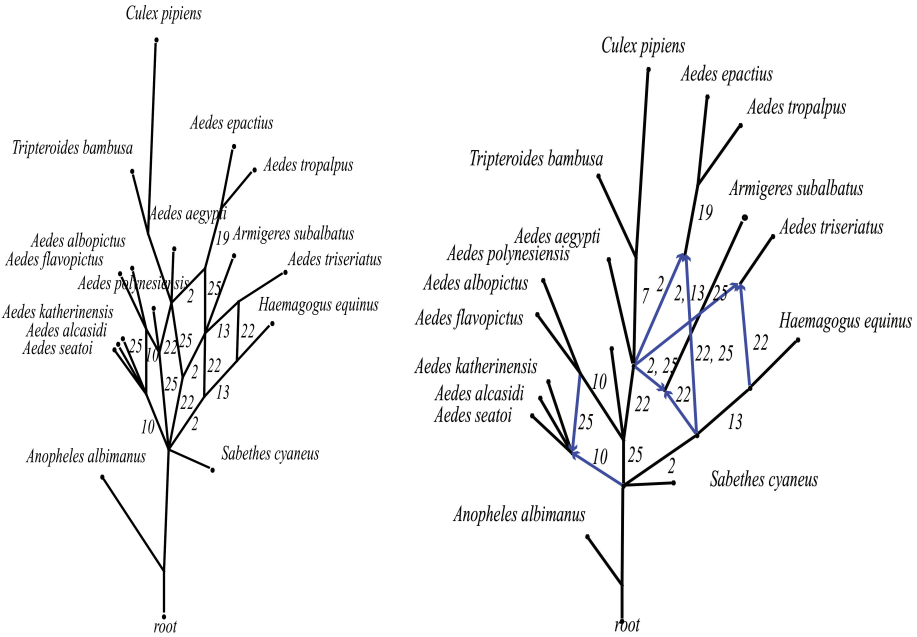
Huson and Klöpper[42] have constructed the split graph and the galled network associated to this dataset (see Fig. 14).

We computed the Hamming distances from the sequences of Table 5 to obtain a distance matrix of size 16 between these species. Then, we applied our algorithm (with DIFFMAX = 0.1) to this matrix and obtained the hybridization network shown in Fig. 15.

We obtained the same number of reticulations (4) as Huson and Klöpper, and the general structure of our network is quite similar to the split graph and galled tree topologies presented in Fig. 14. For example, species *Aedes epactius* and *Aedes atropalpus* are located at the extremity of a reticulation in both networks. However, some significant differences can also be observed. For example, species *Aedes triseriatus* could be considered as a hybrid in the split graph and galled tree topologies,

Table 5. Dataset of the restriction maps of the rDNA cistron of 12 species of mosquitoes constructed using eight recognition restriction enzymes.

Aedes albopictus	1 1 1 1 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0 1 0
Aedes aegypti	1 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 1 0
Aedes seatoi	1 1 1 1 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 0
Aedes flavopictus	1 1 1 1 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0 1 0
Aedes alcasidi	1 1 1 1 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 0
Aedes katherinensis	1 1 1 1 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 0
Aedes polynesiensis	1 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 1 0 1 0 0 1 0
Aedes triseriatus	1 0 1 1 0 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 0 0 0 0 0 0
Aedes atropalpus	1 0 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 1 0
Aedes epactius	1 0 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 1 0
Haemagogus equinus	1 0 1 1 0 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0
Armigeres subalbatus	1 0 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0
Culex pipiens	1 1 1 1 0 1 1 1 0 0 0 1 0 0 0 1 1 1 0 1 0 0 1 0 1 1
Tripteroides bambusa	1 1 1 1 0 1 1 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 1 0
Sabethes cyaneus	1 1 1 1 0 1 0 1 0 0 1 1 0 0 0 1 0 1 0 1 0 1 0 0 0 0
Anopheles albimanus	1 1 0 1 1 1 0 1 1 0 0 1 0 1 1 1 0 1 0 1 1 1 0 1 0 0



(a) Split graph network (b) Recombination galled network

Fig. 14. Split graph and galled network obtained for the rDNA cistron dataset in Table 5.

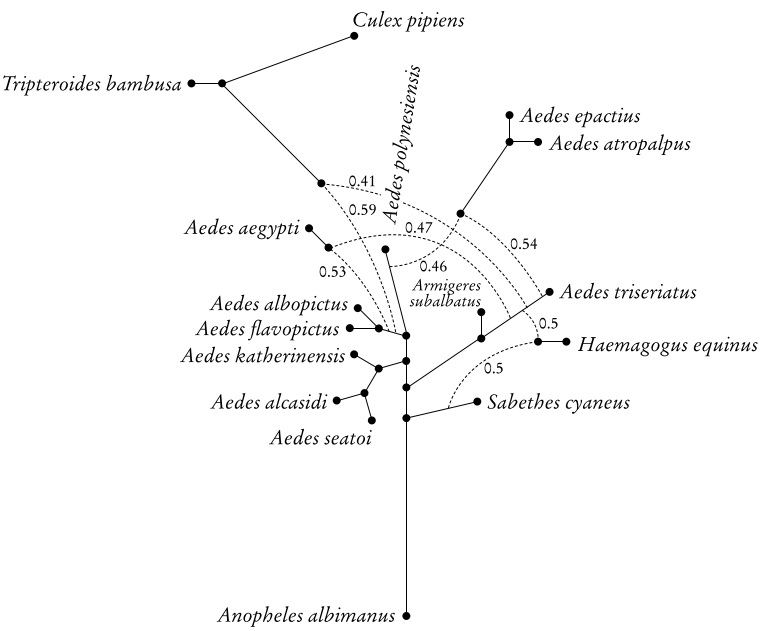


Fig. 15. Hybridation network obtained with our new algorithm. The values of α are indicated on the reticulation branches (depicted by dashed lines).

whereas it is identified as a potential parent of four hybrids in our network. In the same way, species *Sabethes cyaneus* is not involved in any reticulation in the split graph and galled tree topologies, whereas it is a parent of a hybrid in our network. The main advantage of our network representation over split graphs and galled trees is that it identifies hybrids and their parents explicitly. Moreover, our algorithm provides the exact hybridization levels α, and these levels are compatible with the sequences of hybrids and their parents (see Table 5). For example, the sequence of species *Haemagogus equinus* can be obtained by concatenation of the first half of the sequence of species *Aedes triseriatus* and the second half of the sequence of species *Sabethes cyaneus*.

It is worth noting that in this example we changed the condition $\mathrm{MIN}_{i_H,j}^{h_H} > 0$ to $\mathrm{MIN}_{i_H,j}^{h_H} > -0.01$ at step 4 of our algorithm. Indeed, we had $\mathrm{MIN}_{i_H,j}^{h_H} = 0$ for all potential hybrids. This kind of adaptation could be used when the number of detected hybrids is too small (or too large, in the latter case, we should replace 0 by a small positive threshold). This threshold is one of the parameters of our program.

8. Conclusion

We have described a novel fast algorithm for inferring hybridization networks from distance matrices based on the NJ principle. These distance matrices, assumed to encompass contradictory evolutionary signals, could be obtained from the concatenation of genetic sequences or directly from the comparison of genomes of the observed species. The new algorithm provides a good practical solution to the complex problem of the identification of hybridization events. The algorithm's time complexity of $O(n^3)$ makes it applicable for the analysis of large genomic datasets. Moreover, the quality of the obtained results improves as the numbers of considered species grows. The new algorithm finds the exact tree solution when the input distance matrix is a tree metric (or a distance close to a tree metric). The true positive detection rate is very high and the correct hybrids parents are always recovered for additive networks when the hybrids are located between terminal branches. We also provide a way of recovering the correct additive networks when the hybrids are located at any place in the network. The simulation study carried out with sequence data provided very good detection rates as well, even though the false positive rates were a little bit higher in this case.

The execution of our algorithm on the rDNA cistron data[56] allowed us to infer an explicit hybridization network which was compared to the split graph[42] and galled tree[42] topologies. Mention that both split graph and galled tree algorithms infer only implicit phylogenetic networks and are not capable of determining the precise levels of hybridization.

In the future, it would be also important to investigate in more detail how the new technique copes with the tree reconstruction artifacts which generally affect phylogenetic analysis; the main of them are long-branch attraction and unequal evolutionary rates.

The software implementing the discussed algorithm was implemented in the $C++$ language. It is freely available to the research community at the following URL address: http://www.info2.uqam.ca/~makarenkov_v/makarenv/hybrids_detection.zip.

References

1. Saitou N, Nei M, The neighbor-joining method: A new method for reconstructing phylogenetic trees, *Mol Biol Evol* **4**:406–425, 1987.
2. Sneath PHA, Sokal RR, *Numerical Taxonomy. The Principles and Practice of Numerical Classification*, WH Freeman, San Francisco, CA, USA, 1973.
3. Fitch WM, Toward defining the course of evolution: Minimum change for a specific tree topology, *Syst Zool* **4**:406–416, 1971.
4. Felsenstein J, Evolutionary trees from DNA sequences: A maximum likelihood approach, *J Mol Evol* **17**:368–376, 1981.
5. Rannala B, Yang Z, Probability distribution of molecular evolutionary trees: A new method of phylogenetic inference, *J Mol Evol* **43**:304–311, 1996.
6. Sneath PHA, Sackin MJ, Ambler RP, Detecting evolutionary incompatibilities from protein sequences, *Syst Biol* **24**:311–332, 1975.
7. Leclerc B, Makarenkov V, On some relations between 2-trees and tree metrics, *Discrete Math* **192**:223–249, 1998.
8. Makarenkov V, Leclerc B, Comparison of additive trees using circular orders, *J Comput Biol* **7**:731–744, 2000.
9. Sonea S, Panisset M, *A New Bacteriology*, Jones and Bartlett, Burlington, MA, USA, 1983.
10. Stephens JC, Statistical methods of DNA sequence analysis: Detection of intragenic recombination or gene conversion, *Mol Biol Evol* **2**:539–556, 1985.
11. Sawyer S, Statistical tests for detecting gene conversion, *Mol Biol Evol* **6**:526–538, 1989.
12. Hein J, A heuristic method to reconstruct the history of sequences subject to recombination, *J Mol Evol* **36**:396–406, 1993.
13. Rieseberg LH, Morefield JD, Character expression, phylogenetic reconstruction, and the detection of reticulate evolution, in Hoch PC, Stephenson AG (eds.), *Experimental and Molecular Approaches to Plant Biosystematics*, Monographs in Systematic Botany at the Missouri Botanical Garden, Saint-Louis, MO, USA, pp. 333–353, 1995.
14. Bandelt HJ, Dress AWM, A canonical decomposition theory for metrics on a finite set, *Adv Math* **92**:47–105, 1992.
15. Bandelt HJ, Dress AWM, Split decomposition: A new and useful approach to phylogenetic analysis of distance data, *Mol Phylogenet Evol* **1**:242–252, 1992.
16. Bryant D, Moulton V, Neighbor-net: An agglomerative method for the construction of phylogenetic networks, *Mol Biol Evol* **21**:255–265, 2004.
17. Huson DH, Bryant D, Application of phylogenetic networks in evolutionary studies, *Mol Biol Evol* **23**:254–267, 2006.
18. Gambette P, Huson DH, Improved layout of phylogenetic networks, *IEEE/ACM Trans Comput Biol Bioinform* **5**:472–479, 2008.
19. Legendre P, Makarenkov V, Reconstruction of biogeographic and evolutionary networks using reticulograms, *Syst Biol* **51**:199–216, 2002.
20. Makarenkov V, Legendre P, From a phylogenetic tree to a reticulated network, *J Comput Biol* **11**:195–212, 2004.
21. Makarenkov V, Legendre P, Improving the additive tree representation of a given dissimilarity matrix using reticulations, in Kiers HAL, Rasson JP, Groenen PJF, Schader M (eds.), *Data Analysis, Classification, and Related Methods, Proc. 7th Conf. Int.*

Federation of Classification Societies (*IFCS-2000*), Springer Verlag, Berlin/Heidelberg, Germany, pp. 35–40, 2000.
22. Hallett MT, Lagergren J, Efficient algorithms for lateral gene transfer problems, in *RECOMB'01: Proc. 5th Annual Int. Conf. Computational Biology*, ACM, New York, NY, USA, pp. 149–156, 2001.
23. Makarenkov V, Boc A, Delwiche CF, Diallo AB, Philippe H, New efficient algorithm for modeling partial and complete gene transfer scenarios, in *Data Science and Classification*, Springer, Berlin/Heidelberg, Germany, pp. 341–349, 2006.
24. Boc A, Philippe H, Makarenkov V, Inferring and validating horizontal gene transfer events using bipartition dissimilarity, *Syst Biol* **59**:195–211, 2010.
25. Boc A, Makarenkov V, Towards an accurate identification of mosaic genes and partial horizontal gene transfers, *Nucleic Acids Res* **39**:e144, 2011.
26. Makarenkov V, T-REX: Reconstructing and visualizing phylogenetic trees and reticulation networks, *Bioinformatics* **17**:664–668, 2001.
27. Boc A, Diallo AB, Makarenkov V, T-REX: A web server for inferring, validating and visualizing phylogenetic trees and networks, *Nucleic Acids Res* **40**:W573–W579, 2012.
28. Bandelt HJ, Forster P, Röhl A, Median-joining networks for inferring intraspecific phylogenies, *Mol Biol Evol* **16**:37–48, 1999.
29. Doyon JP, Scornavacca C, Gorbunov KY, Szöllosi GJ, Ranwez V, Berry V, An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers, in Tannier E (ed.), *Comparative Genomic*, Springer, Berlin/Heidelberg, Germany, pp. 93–108, 2010.
30. Huson DH, Rupp R, Summarizing multiple gene trees using cluster networks, in Crandall KA, Lagergren J (eds.), *Algorithms in Bioinformatics*, Springer, Berlin/Heidelberg, Germany, pp. 296–305, 2008.
31. van Iersel L, Kelk S, Rupp R, Huson D, Phylogenetic networks do not need to be complex: Using fewer reticulations to represent conflicting clusters, *Bioinformatics* **26**:i124–i131, 2010.
32. Albrecht B, Scornavacca C, Cenci A, Huson DH, Fast computation of minimum hybridization networks, *Bioinformatics* **28**:191–197, 2012.
33. Chen Z-Z, Wang L, Yamanaka S, A fast tool for minimum hybridization networks, *BMC Bioinformatics* **13**:155, 2012.
34. Wu Y, Close lower and upper bounds for the minimum reticulate network of multiple phylogenetic trees, *Bioinformatics* [*ISMB*] **26**:140–148, 2010.
35. Chen Z-Z, Wang L, Algorithms for reticulate networks of multiple phylogenetic trees, *IEEE/ACM TCBB* **9**:372–384, 2012.
36. van Iersel L, Kelk S, Constructing the simplest possible phylogenetic network from triplets, *Algorithmica* **60**:207–235, 2011.
37. Schmidt HA, Strimmer K, Vingron M, von Haeseler A, TREE-PUZZLE: Maximum likelihood phylogenetic analysis using quartets and parallel computing, *Bioinformatics* **18**:502–504, 2002.
38. Strimmer V, von Haeseler A, Quartet puzzling: A quartet maximum-likelihood method for reconstructing tree topologies, *Mol Biol Evol* **13**:964–969, 1996.
39. Grünewald S, Forslund K, Dress A, Moulton V, QNet: An agglomerative method for the construction of phylogenetic networks from weighted quartets, *Mol Biol Evol* **24**:532–538, 2007.
40. Yang J, Grünewald S, Wan XF, Quartet-Net: A quartet based method to reconstruct phylogenetic networks, *Mol Biol Evol* **30**:1206–1217, 2013.
41. Huson DH, Klöpper TH, Computing recombination networks from binary sequences, *Bioinformatics* **21**:ii159–ii165, 2005.

42. Huson DH, Klöpper TH, Beyond galled trees — decomposition and computation of galled networks, in Speed TP, Huang H (eds.), *Research in Computational Molecular Biology*, Springer, Berlin/Heidelberg, Germany, pp. 211–225, 2007.
43. Woolley SM, Posada D, Crandall KA, A comparison of phylogenetic network methods using computer simulation, *PLoS ONE* **3**:e1913, 2008.
44. Semple C, Hybridization networks, in Gascuel O, Steel M (eds.), *Reconstructing Evolution: New Mathematical and Computational Advances*, Oxford University Press, Oxford, UK, pp. 277–314, 2007.
45. Stace CA, *Plant Taxonomy and Biosystematics*, Cambridge University Press, Cambridge, UK, 1991.
46. Judd WS, *Plant Systematics: A Phylogenetic Approach*, Sinauer Associates, Sunderland, MA, USA, 2008.
47. Dawley RM, An introduction to unisexual vertebrates, in Dawley RM, Bogart JP (eds.), *Evolution and Ecology of Unisexual Vertebrates*, New York State Museum, Albany, NY, USA, pp. 1–18, 1989.
48. Barthélemy JP, Guénoche A, *Trees and Proximity Representations*, John Wiley & Sons, Hoboken, NJ, USA, 1991.
49. Buneman P, A note on metric properties of trees, *J Comb Theory* **17**:48–50, 1974.
50. Atteson K, The performance of neighbor-joining methods of phylogenetic reconstruction, *Algorithmica* **25**:251–278, 1999.
51. Studier JA, Keppler KJ, A note on the neighbor-joining algorithm of Saitou and Nei, *Mol Biol Evol* **5**:729–731, 1988.
52. Kuhner M, Felsenstein J, A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates, *Mol Biol Evol* **11**:459–468, 1994.
53. Rambaut A, Grass NC, Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees, *Comput Appl Biosci* **13**:235–238, 1997.
54. Kimura M, A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences, *J Mol Evol* **16**:111–120, 1980.
55. Felsenstein J, PHYLIP — Phylogeny Inference Package (Version 3.2), *Cladistics* **5**:164–166, 1989.
56. Kumar A, Black WC, Rai KS, An estimate of phylogenetic relationships among culicine mosquitoes using a restriction map of the rDNA cistron, *Insect Mol Biol* **7**:367–373, 1998.

Matthieu Willems received his Doctoral degree in mathematics from the Université Paris Diderot (Paris 7) in 2003. He has held postdoctoral fellowships at the University of Toronto, McGill University and the University of Ottawa. He is currently pursuing a doctoral degree in bioinformatics at the Université du Québec à Montréal. His current research focuses on phylogenetic networks.

Nadia Tahiri is a Ph.D. student in Computer Science at the Université du Québec à Montréal. She works on the development of new bioinformatics consensus algorithms.

Vladimir Makarenkov is professor at the Department of Computer Science at the Université du Québec à Montréal. His research interests are in the fields of Bioinformatics, Operations Research, Software Engineering, and Mathematical Classification.