Laboratoire d'Informatique, de Robotique
et de Microélectronique de Montpellier

Unité Mixte CNRS - Université Montpellier II C 09928

RAPPORT DE RECHERCHE

Inferring Evolutionary Trees with Strong Combinatorial Evidence

V. Berry & O. Gascuel

R.R.LIRMM 97-032

161, Rue Ada - 34392 Montpellier Cedex 5 - France
Tél: (33) 67 41 85 85 - Fax: (33) 67 41 85 00

Inferring Evolutionary Trees with Strong Combinatorial Evidence

Vincent Berry* Olivier Gascuel*†

Abstract

We consider the problem of inferring evolutionary trees. We propose a quartet reconstruction method which specifically produces trees whose edges have strong combinatorial evidence. For this purpose we use the Q^* relation introduced by Bandelt and Dress [4], which they demonstrated to be definable as the maximum subset of resolved quartets which is equivalent to a tree. We further investigate the properties of this variation of the NP-hard quartet consistency problem, first providing a polynomial time, $O(n^4)$, algorithm, where n is the number of species. Moreover, we show that the convergence rate of the method, under the Cavender-Farris model of evolution, is polynomial for realistic conditions. Last, we exemplify the method on a set of mammal DNA sequences.

1 Introduction

A fundamental problem in computational biology is to retrieve the history of a set of species by reconstructing their evolutionary tree. Such a tree, also called a *phylogeny*, has its leaves bijectively labelled by the studied species, while internal nodes represent hypothetical ancestors. Evolutionary data used to reconstruct a phylogeny often consists in homologous DNA sequences issued from the species' genome. These data are sometimes translated into a matrix of pairwise distances between species, corrected according to a given model of evolution, to account for hidden mutation events. A complete survey on phylogenetic reconstruction criteria and algorithms can be found in [30].

Recently there has been strong interest in providing polynomial time algorithms with performance guarantees. Agarwala *et al* [1] proposed a 3-approximation algorithm for the L_∞-nearest tree problem. Then, Farach

*Département d'Informatique Fondamentale, LIRMM, Université de Montpellier II, France.

†GERAD, 5255 Avenue Decelles, Montreal (Quebec) H3T1V6 - Canada.

and Kannan [14] first proved a guaranteed convergence rate for a reconstruction method. This result is of importance in the phylogenetic domain since the amount of available data (*i.e.*, the sequence length) is the critical resource. More recently, Erdös *et al* [13] and Atteson [3] improved this result by proving faster convergence rates, for a new quartet method and for the well-known NJ heuristics [26], respectively.

We investigate here a reconstruction method which specifically produces trees with edges showing strong combinatorial evidence, and for which we can also prove the convergence rate. It is based on a quaternary relation introduced by Bandelt and Dress [4], and thus relies on a quartet reconstruction principle [4, 10, 13, 27, 28, 29]. Quartet methods first compute the topologies of subtrees induced by subsets of four species. Then, they rely on a combinatorial algorithm to construct a tree on the entire set of species, respecting as many of the topological constraints imposed by the 4-species subtrees as possible. The interest of this approach is that many methods exist which can efficiently infer topologies on four species, but which can not be applied to many more species due to mathematical or computational difficulties. Thus the best (or only) way to use these methods for phylogenetic inference on larger sets of data, is to combine the *resolved* 4-species trees (or *r4-trees*) they infer.

Any phylogeny T is uniquely determined by its set of r4-trees [10], from which it can be reconstructed in polynomial time [4]. In contrast, knowing if a given r4-tree set Q is *consistent* with a tree, *i.e.*, if there exists a tree respecting all topological constraints imposed by Q, is NP-Complete [28]. Several heuristics have been designed to solve the corresponding optimization problem or some of its variations [4, 12, 13, 29]. We consider here the variation introduced in [4], *i.e.*, we are interested in the quaternary relation Q^* corresponding to the maximum subset of Q which is *tree-like*. In other words, Q^* is consistent with a tree, noted T^*, which in turn only induces r4-trees present in Q^*. Independently of the way the set Q is obtained, the previous characterization makes T^* a suitable tree representation of the data, since it neither induces any contradiction with them, nor necessitates any new hypothesis. Moreover, there is no tree-consistent subset of Q larger than Q^* which corresponds to a tree with these properties. T^* can be seen as a combinatorially safe estimate of the unknown phylogeny due to the stringent combinatorial constraints imposed on its non-trivial edges, *i.e.*, each one must respect between $O(n^2)$ and $O(n^4)$ data r4-trees. In this way, no wrong edges are likely to be inferred due to random errors made when sampling the data, inferring the r4-trees, or choosing edges of the tree.

By producing only trees with edges showing convincing evidence, we will often infer partially resolved trees (with some nodes of degree >3), as also proposed recently in [23]. This approach has the advantage of mainly avoiding the overfitting effect from which usual reconstruction methods suffer. Indeed, these methods usually produce fully resolved trees (*i.e.*, with degree 3 internal nodes only), providing a complete explanation of the inter-species relationships. However, it is known that we can rarely recover accurately the entire topology of the unknown phylogeny, given sequences of realistic length. The best evidence for this phenomenon is the high variability observed within trees obtained by different methods on the same piece of data [15, 20], or when slightly varying the set of studied species [21, 25].

For the various reasons given above, it appears that T^* would be useful for phylogenetic reconstruction - if we could compute it efficiently.

Our results

In their paper, Bandelt and Dress gave no algorithmic guarantee for this problem but they thought (Bandelt, personal communication) that a polynomial time algorithm would be achievable. We show here that Q^* can actually be computed in polynomial time, solving this variation of the NP-Hard quartet consistency problem. The algorithm we propose, IQ^*, relies on an *incremental* principle: starting from four species, T^* is built up by progressively attaching the remaining species to the tree. At each step, the introduced species may lead to new inter-species relationships (internal edges) but also to remove previous relationships which can not be extended to account for it. With this principle we obtain an $O(n^4)$ algorithm, whose analysis would indicate a small leading coefficient (as confirmed by simulations). The same complexity is also usually required to infer the r4-trees from biological data, so that the whole phylogenetic reconstruction method based on IQ^* is in $O(n^4)$. Note also that the complexity bound that we obtain for the optimization problem (*i.e.*, computing Q^*) is the same as that required by the best algorithm [4], to our knowledge, which solves the corresponding decision problem (*i.e.*, deciding whether Q is tree-like).

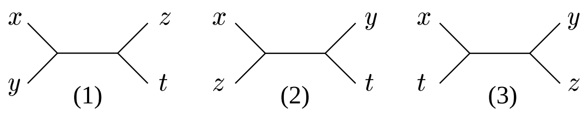
Moreover, using a recent result of Erdös *et al* [13], we show that the IQ^* algorithm, associated with a distance-based r4-tree inference method, has a guaranteed polynomial convergence rate under realistic conditions and when assuming the Cavender-Farris model of evolution [9, 13, 14].

In the following, we first give prerequisites and define precisely the Q^* relation (section 2). Then we concentrate on properties which are important for applying Q^* to phylogenetic reconstruction, first proving the incremental principle (section 3), then giving the $O(n^4)$ algorithm (section 4), and next showing its convergence rate (section 5). We conclude by an example on real data.

2 Reconstructing phylogenies from r4-trees

Definition 2.1 *A* phylogeny *for a set $S = \{1, 2, \ldots, n\}$ of species is a tree whose leaves are bijectively labelled by the species of S and whose internal (i.e., non-leaf) nodes have degree ≥ 3.*

To any quartet of species $\{x, y, z, t\}$ there are four ways to associate a tree. The three possible topologies with ternary internal nodes (Figure 1) are noted $xy|zt$, $xz|yt$ and $xt|yz$, indicating how the species are split into two pairs by the central edge (note that $xy|zt \equiv yx|zt \equiv zt|yx$).



r4-tree $xy|zt$ r4-tree $xz|yt$ r4-tree $xt|yz$

Figure 1: Resolved topologies on 4 species

These topologies are called *r4-trees* (for *resolved 4-trees*). They can be viewed as topological constraints that the phylogeny on the entire set of species must respect. The star topology (with one internal node) is not an r4-tree since it represents here an absence of decision for the quartet of species rather than a topological constraint.

Any phylogeny T can be characterized by its r4-trees: T *induces* the r4-tree $xy|zt$ iff the paths $[xy]$ and $[zt]$ are distinct in T. In this case, the topologies of both this r4-tree and the subtree of T split the four species in the same way, each edge of the r4-tree possibly corresponding to several edges in T. The case where the paths $[xy]$ and $[zt]$ intersect in just one node of T corresponds to the star topology, thus T induces no r4-tree for the corresponding quartet. This situation arises whenever T has some nodes of degree ≥ 4. Let Q_T denote the set of r4-trees induced by T. It is easy to see that Q_T contains at most one r4-tree for each quartet of species. Similarly, we will only consider arbitrary r4-tree sets Q containing at most one r4-tree for each quartet of species (as is most usually the case in practice).

Definition 2.2 *An r4-tree set Q is* tree-consistent *iff $\exists\, T / Q \subseteq Q_T$. Q is* tree-like *iff $\exists\, T / Q_T = Q$.*

A phylogeny T can also be characterized by the set of bipartitions, or *splits*, its edges induce on the set S of *all* species [4, 7]. Deleting any edge from T disconnects T into two components, and thereby induces a bipartition on S consisting of the species of the two components. To each bipartition $b = \sigma|\overline{\sigma}$ we associate the set $Q_b = \{xy|zt / x, y \in \sigma,\ z, t \in \overline{\sigma}\}$ of r4-trees it *induces*. A bipartition is called *trivial* when one of its components contains less than 2 species, and hence induces no r4-tree. A set of bipartitions is *tree-like* iff there exists a tree with edges corresponding to its bipartitions. We clearly see that there is a straightforward containment relation between the concepts of r4-tree, bipartition and tree: a tree may be considered as a set of bipartitions and a bipartition as a set of r4-trees.

Bandelt and Dress [4] introduced the subset Q^* of a given r4-tree set Q in the following way:

DEFINITION 2.3 *Let B^* be the set of bipartitions $b = \sigma|\overline{\sigma}$ such that $Q_b \subseteq Q$, then $Q^* = \bigcup_{b \in B^*} Q_b$.*

They proved (proposition 3) that this definition of Q^* corresponds to the unique maximum subset of Q which is tree-like. We denote by T^* the tree which corresponds to Q^* (and B^*). Obtaining T^*, Q^* or B^* from one another is easy, *e.g.*, B^* gives T^* in $O(n)$ [19, 22].

If Q is tree-like, then $Q^* = Q$ and T^* is the corresponding tree. If Q is not tree-like, then in the worst case we have $Q^* = \emptyset$, thus all bipartitions of B^* are trivial and T^* corresponds to the star topology on S.

3 Computing Q^* by an incremental principle

In [4], Bandelt and Dress did not provide any algorithm for computing Q^*. However, in another context sharing similarities with that of Q^*, they mention an $O(n^6)$ incremental algorithm to obtain the $O(n^2)$ *d-splits* of a distance matrix d [5]. When applying the same principle to Q, we derive an $O(n^5)$ algorithm to obtain the $O(n)$ splits of T^*. We first show below the correctness of the incremental approach for computing $B^* \equiv Q^*$ and give a simple algorithm which results. In the next section we show how the complexity bound can be reduced down to $O(n^4)$.

DEFINITION 3.1 *We suppose an arbitrary order $(1, 2, \ldots, n)$ on the species of S and let $[i]$ denote the first i species.*

- *Q_i denotes the subset of r4-trees $xy|zi$ of Q referencing i and such that $\{x, y, z\} \subseteq [i-1]$.*
- *$Q_{[i]} = \bigcup_{j \leq i} Q_j$ denotes the subset of r4-trees only referencing species in $[i]$.*
- *B_i^* is the set of bipartitions $b = \sigma|\overline{\sigma}$ on $[i]$ such that $Q_b \subseteq Q_{[i]}$. Note that B_i^* contains at least the $i+1$ trivial bipartitions $\emptyset|\{1, \ldots, i\}$, $\{1\}|\{2, \ldots, i\}$, $\ldots$, $\{i\}|\{1, \ldots, i-1\}$.*
- *Q_i^* denotes the set of r4-trees $xy|zt \in Q_{[i]}$ such that $\exists\, b \in B_i^*$, $xy|zt \in Q_b$.*

From the above definitions, we see that Q_i^* is the unique maximum subset of $Q_{[i]}$ which is tree-like (T_i^* denotes the corresponding tree) and we have $Q^* = Q_n^*$.

The incremental principle consists in focusing on the successive bipartition sets B_i^* from which the r4-tree sets Q_i^* are defined. At each step a new species (let i) is considered and the set B_i^* is obtained from B_{i-1}^*. Each bipartition $\sigma|\overline{\sigma} \in B_{i-1}^*$, defined on $[i-1]$, may *a priori* be extended into two bipartitions defined on $[i]$: $b = \sigma \cup \{i\}|\overline{\sigma}$ and $b' = \sigma|\overline{\sigma} \cup \{i\}$. B_i^* contains one, both or neither of them, depending on $Q_b \subseteq Q$ and $Q_{b'} \subseteq Q$.

Our next step is to specify the relation connecting the sets B_{i-1}^* and B_i^*.

3.1 Correctness of the approach

DEFINITION 3.2 *Let b be a bipartition on $[i]$, $b_{\backslash i}$ denotes its* restriction *to the species $[i-1]$. In a wider sense, $B_{\backslash i}^*$ denotes the bipartitions of B_i^* restricted to $[i-1]$.*

For example, if $b = \{1,2,3\}|\{4,5,6\}$, then $b_{\backslash 6} = \{1,2,3\}|\{4,5\}$.

LEMMA 3.1 $B_{\backslash i}^* \subseteq B_{i-1}^*$

Proof. Let $b = \sigma|\overline{\sigma}$ be a bipartition of B_i^*. If b is trivial then $b_{\backslash i}$ is trivial and thus belongs to B_{i-1}^*. Otherwise assume that $i \in \sigma$ (equiv. $i \in \overline{\sigma}$). $b \in B_i^*$ implies $\forall x, y \in \sigma$ and $z, t \in \overline{\sigma}$, $\{x,y,z,t\} \subseteq [i]$ and $xy|zt \in Q_{[i]}$. Thus, either (1) $b_{\backslash i} = \sigma - \{i\}|\overline{\sigma}$ is trivial ($|\sigma - \{i\}| = 1$) implying $b_{\backslash i} \in B_{i-1}^*$, or (2) $\forall x, y \in \sigma - \{i\}$ and $z, t \in \overline{\sigma}$ we have $\{x,y,z,t\} \subseteq [i-1]$ and $xy|zt \in Q_{[i]}$, thus more precisely $xy|zt \in Q_{[i-1]}$, implying $b_{\backslash i} \in B_{i-1}^*$. □

The lemma can not be extended to state that $B_{\backslash i}^* = B_{i-1}^*$, as the following example shows: let $Q = \{12|34, 15|23, 15|24, 15|34\}$, we have $b = \{1,2\}|\{3,4\} \in B_4^*$, and yet $b \notin B_{\backslash 5}^*$, since, *e.g.*, $25|34 \notin Q$ and $12|35 \notin Q$.

COROLLARY 3.1 *Any bipartition of B_i^* may be directly obtained from B_{i-1}^*, i.e., $\forall b \in B_i^*$, $\exists\, b' = \sigma|\overline{\sigma} \in B_{i-1}^* \,/\, b = \sigma \cup \{i\}|\overline{\sigma}$ or $b = \sigma|\overline{\sigma} \cup \{i\}$.*

Moreover, let b be a bipartition obtained by extending a bipartition $b' \in B_{i-1}^*$. To know if $b \in B_i^*$, we do not have to check for the presence of all r4-trees of Q_b in Q, since we already know $Q_{b'} \subseteq Q$. As $Q_{b'} \subseteq Q_b$, we only have to examine the new r4-trees ($Q_b - Q_{b'}$), induced by the addition of i in $b' = \sigma|\overline{\sigma}$: to know if $b = \sigma \cup \{i\}|\overline{\sigma}$ (or equiv. $b = \sigma|\overline{\sigma} \cup \{i\}$) may be derived, we only check whether the subset Q_i of Q contains the set of r4-trees $xi|yz$ for which $x \in \sigma$ and $y, z \in \overline{\sigma}$.

3.2 A simple incremental algorithm

The algorithm first considers the trivial case of four species, for which B_4^* is readily obtained: it contains the five trivial bipartitions on these species plus possibly the bipartition $\{x,y\}|\{z,t\}$ (if $xy|zt \in Q$, with $\{x,y,z,t\} = \{1,2,3,4\}$). Then, the algorithm progressively enlarges the set of considered species and consequently extends the set of bipartitions. At the step where species i is considered, for each bipartition $b = \sigma|\overline{\sigma} \in B^*_{i-1}$, we must check whether $b' = \sigma\cup\{i\}|\overline{\sigma}$ and $b'' = \sigma|\overline{\sigma}\cup\{i\}$ qualify as bipartitions of B^*_i. As stated before, this only requires checking whether $(Q_{b'} - Q_b) \subseteq Q_i$ and $(Q_{b''} - Q_b) \subseteq Q_i$, respectively. This procedure stops after species n has been processed.

This simple algorithm computes B^* in $O(n)$ steps. Each step examines the two possible extensions of $O(n)$ bipartitions, each time potentially examining the $O(n^3)$ r4-trees referencing i, which gives an $O(n^5)$ algorithm. In fact, a finer analysis shows that the complexity oscillates between $O(n^5)$ for *caterpillar* trees (as in Figure 2) and $O(n^4 \log n)$ for well balanced trees.

4 IQ^*: an $O(n^4)$ algorithm

We now present the IQ^* algorithm which consists in an improved version of the previous simple algorithm. IQ^* is based on the same incremental principle, but it enables us to obtain B^*_i in time $O(n^3)$ at each step rather than $O(n^4)$ previously. The main idea with respect to the previous algorithm is to share information between the *components* of B^*_{i-1} bipartitions on the basis of the r4-trees they commonly concern.

DEFINITION 4.1 *Let i be the species currently considered and $b = \sigma|\overline{\sigma}$ a bipartition on $[i-1]$; we say that an r4-tree $xi|yz$ is* required *(to be in Q) by the component σ of b if it is induced by the addition of i to this component, i.e., if $x \in \sigma$ and $y, z \in \overline{\sigma}$. An r4-tree is called* processed *when its presence in Q has been examined. In the following, the term* subtrees *only denotes the connected component of a tree T obtained by deleting one of its edges (but not its endpoints).*

Note that the above definition only considers newly required r4-trees (*i.e.*, referencing the species i) since other r4-trees need not be reexamined as stated in the previous section.

Each r4-tree may be required by components of several bipartitions. For example, in Figure 2, the r4-tree $xi|yz$ is required by the component σ_1 of the bipartition $\sigma_1|\overline{\sigma}_1$ but also by the component σ_2 of $\sigma_2|\overline{\sigma}_2$ since adding i to both these components, *i.e.*, inserting i to the left of e_1, would lead the tree to induce the r4-tree $xi|yz$.

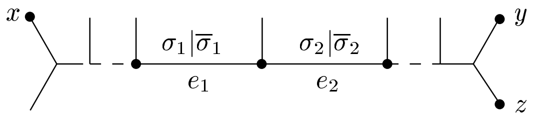


Figure 2:

How do we know all the components that require a given r4-tree? Simply, in taking advantage of the containment relations existing between components. For example in Figure 2, $x \in \sigma_1$ and $x \in \sigma_2$, thus by definition, all r4-trees $xi|yz$ which reference pairs of species $y, z \in \overline{\sigma}_2$ are required by component σ_2, but as $\overline{\sigma}_2 \subseteq \overline{\sigma}_1$, they are also required by σ_1.

To systematize the use of these containment relations in the algorithm, we need to know easily all such relations between components of $\mathcal{B}^*_{i-1}$ bipartitions. Since these bipartitions are consistent with a tree (T^*_{i-1}), their components are in one to one correspondence with the subtrees of this tree, as is well-known [4, 7]. Thus we only have to follow T^*_{i-1} topology to obtain the containment relations between components from the ones induced between subtrees. For example, in Figure 2 every internal edge on e_1's right has its right component (subtree) contained in e_1's right component (subtree), whereas its left component contains e_1's left component. The symmetric may also be derived for edges on e_1's left.

Figure 3 illustrates the situation arising around any given internal node r ($\sigma, \overline{\sigma}, \sigma_j, \overline{\sigma}_j, j \in 1..l$ will denote components or associated subtrees depending on the context). The r4-trees $xi|yz$ required by any $\sigma_j, j \in 1..l$ and such that $x \in \sigma$, are also required by $\sigma \subset \sigma_j$.

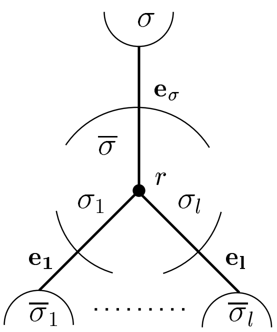


Figure 3: Containment relations on components.

We then easily see the interest lying in an ordered processing of the components. Suppose the components $\sigma_1, \dots, \sigma_l$ are processed before σ.

The information of whether the species i can be added to these components could be used in order to know if i can be added to σ, without testing Q again for the presence of the r4-trees they commonly require. More generally, any component which contains σ shares some required r4-trees with it. This indicates that all *supertrees* of σ should be processed before σ is considered, calling naturally for the use of a recursive search of the tree T^*_{i-1}, since containment relations between components follow descendant relations between subtrees. We will later detail how a suitable processing order on the components may be determined this way. However, this principle may be of some use only if we can easily characterize the required r4-trees that remain to be processed for a component, when all components that contain it have already been considered. We solve this argument below.

For any bipartition $\sigma|\overline{\sigma}$ of B^*_{i-1}, in the following we only consider *w.l.o.g.* the components σ_j which *minimally* contain σ (*i.e.*, $\sigma_j \supset \sigma$ and $\nexists \theta \,/\, \sigma_j \supset \theta \supset \sigma$), and their *opposite* components, $\overline{\sigma}_j$, *maximally* contained in $\overline{\sigma}$ ($\overline{\sigma}_j \subset \overline{\sigma}$ and $\nexists \theta \,/\, \overline{\sigma}_j \subset \theta \subset \overline{\sigma}$). Then,

Definition 4.2 *Let R_σ denote the set of all r4-trees required by σ, then $R^c_\sigma (\subseteq R_\sigma) = \bigcup_{j,k} R^{j,k}_\sigma$ ($j \neq k \in 1..l$), where $R^{j,k}_\sigma = \{xi|yz \in R_\sigma \,/\, x \in \sigma,\, y \in \overline{\sigma}_j,\, z \in \overline{\sigma}_k\}$, denotes the* crossed *r4-trees of R_σ.*

Claim 4.1 *If components $\sigma_1, \dots, \sigma_l$ have been considered before σ, then all r4-trees of R_σ have been processed except those of R^c_σ.*

Thus, when considering a component σ to know if it can support the addition of species i, we have only to check Q for the presence of its crossed r4-trees R^c_σ. We can then characterize the recursive process of the components of B^*_{i-1} bipartitions as follows: *a)* Always consider a component *after* the components that contain it; *b)* At the step where a component σ is considered, decide whether it can be extended to include the species i according to $R^c_\sigma \subseteq Q$ *and* to the information derived for other required r4-trees of R_σ, examined during previous steps. According to these two points, it remains to be specified: *a)* How do we obtain a processing order of the components insuring that each component be considered after the components which contains it; *b)* How do we pass information between components, to indicate the result of the previous processing of r4-trees they commonly require.

4.1 Transmitting information

As previously seen, if $\sigma \subset \sigma_j$, then some r4-trees required by σ_j are also required by σ. However, not all r4-trees of R_{σ_j} are relevant for σ. More precisely, r4-trees $xi|yz \in R_{\sigma_j}$ such that $x \in \sigma_j - \sigma$ (*i.e.*, $x \in \overline{\sigma}_k, j \neq k$) are not required by σ since $x \notin \sigma$. As a consequence, when σ_j can not be extended we can not systematically deduce the same answer for σ, since the negative answer we got for σ_j might just result from r4-trees not relevant for σ. Let $L_j = \{x \,/\, \exists\, xi|yz \in R_{\sigma_j} - Q\}$ denote the set of all species x referenced in an r4-tree required by σ_j but *lacking* in Q. We have $L_j \neq \emptyset$ *iff* σ_j can not be extended. But only when $L_j \cap \sigma \neq \emptyset$, can we deduce from this result that σ can not be extended. This implies that when considering σ we must know the sets L_j for the components σ_j in which it is contained. Note that these sets are available since, according to the principle of the recursive process, all components that contain σ are previously considered. Thus, we have the following decision rule:

CLAIM 4.2 *Component σ can be extended to include the species i iff $\bigcup_j L_j \cap \sigma = \emptyset$ and $R^c_\sigma \subseteq Q$.*

Note that the paradoxal situation may arise where all components containing σ can be extended and yet σ can not, since the negative answer for σ may be due only to the lack of some r4-trees R^c_σ in Q. In terms of Figure 3, this particular case arises when the new external edge connecting i to the tree is attached to node r.

Another particular situation arises when one (or several) component σ_j can not be extended to include i, but $\sigma \subset \sigma_j$ can. When considering this situation from the subtree standpoint, this again seems paradoxal. However, as remarked in section 3, some bipartitions of $\mathcal{B}^*_{i-1}$ will not be extended into any bipartition of B^*_i, meaning that the corresponding edges of the tree will be deleted (by joining their two end-points). This is the case here for the bipartition $\sigma_j|\overline{\sigma}_j$, and removing the corresponding edge of the tree resolves the apparent contradiction.

Algorithm 1 gives the operations performed when considering a component σ, assuming that components $\sigma_1, \dots, \sigma_l$ which contain it have already been processed.

Algorithm 1: Deciding whether a component σ of a B^*_i bipartition can be extended to include the species i

$L_\sigma \leftarrow \bigcup_j L_j \cap \sigma$
foreach *pair* $\overline{\sigma}_j, \overline{\sigma}_k$ *of maximal subtrees of* $\overline{\sigma}$
  foreach $x \in \sigma$, $y \in \overline{\sigma}_j$, $z \in \overline{\sigma}_k$ $(\equiv xi|yz \in R^c_\sigma)$
    if $xi|yz \notin Q$ **then** $L_\sigma \leftarrow L_\sigma \cup \{x\}$

if $L_\sigma = \emptyset$ **then** answer `YES` **else** answer `NO`

The set L_σ is computed as a by-product and will be used by the components considered in future steps. As a consequence, even if we know from the union of the L_j's that i can not be inserted in σ, we still have to examine all r4-trees of R^c_σ for L_σ to be complete. It can be easily checked that $L_\sigma = \emptyset$ whenever $\overline{\sigma}$ is a leaf (corresponding to a species w). This means that any trivial bipartition $\{w\}|\overline{\sigma}$ may always be extended into the bipartition $\{w\}|\overline{\sigma}\cup\{i\}$ of B^*_i (and also possibly into $\{w,i\}|\overline{\sigma}$). Thus T^*_n will contain at least the edges corresponding to the star topology. The trivial bipartition $\emptyset|[i-1]$ of $\mathcal{B}^*_{i-1}$ is somewhat special since it does not correspond to any edge of T^*_{i-1}; it is systematically extended into bipartitions $\emptyset|[i]$ and $\{i\}|[i-1]$.

4.2 Exploring the components

We now consider the problem of determining a processing order of the components of the B^*_{i-1} bipartitions. This order must respect the constraint that a component be processed only after the components which contain it. To satisfy this constraint, the two components of a same bipartition can not generally be considered successively. This implies that several searches of the tree T^*_{i-1} are needed. In the following we prove that two recursive searches are enough.

To simplify the presentation, we consider *w.l.o.g.* a simpler problem, where the same ordering constraint is imposed. Suppose that for all subtrees of a given tree, we must compute the list of the species they each contain, given that a subtree must be considered after the subtrees which contain it. Thus, we know the species associated to a given subtree by intersecting the species sets of the subtrees containing it.

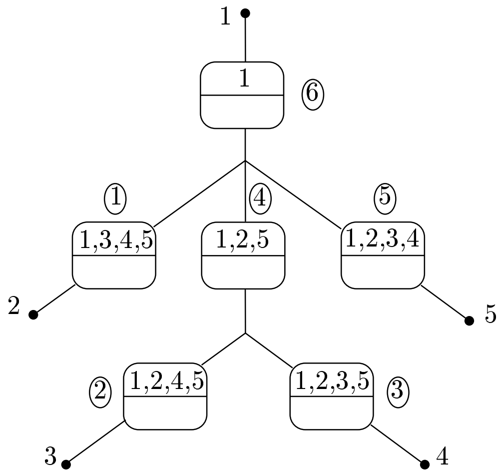


Figure 4: Postordered depth-first search (as shown by circled numbers). Information is passed for the upper subtrees from the leaves to the root.

Choosing an arbitrary leaf of the tree as a root enables us to direct the tree. A *lower* and an *upper* subtree are now associated to the end-points of each edge. The first recursive search of the tree, implemented as a *postordered depth-first search*, only computes the information associated to the upper subtrees, using for each one the information already obtained for its *children* subtrees. *E.g.*, in Figure 4, we know that the upper subtree induced by edge number ④contains the species $\{1, 2, 4, 5\} \cap \{1, 2, 3, 5\}$. Upper subtrees whose opposite (lower) subtree is a leaf are particular cases: they simply contain all species except the one referenced by that leaf (*e.g.*, the upper subtree defined by edge ①, connecting species 2 to the tree, is labelled $\{1, 3, 4, 5\}$. Figure 4 shows the processing order of the edges and the labels obtained for the upper subtrees.

The second recursive search of the tree is implemented as a *preordered depth-first search* in order to process the lower subtrees (Figure 5). The information for each of these subtrees is again computed from the information obtained for the subtrees which contain it. Each lower subtree defined by an edge e is contained in a lower subtree (defined by the *father* edge of e) and also in upper subtrees (the ones defined by *brother* edges of e). *E.g.*, the lower subtree defined by edge number ③ (previously numbered ④) is labelled $\{2, 3, 4, 5\} \cap \{1, 3, 4, 5\} \cap \{1, 2, 3, 4\}$. Here the only particular case is the lower subtree whose opposite (upper) subtree corresponds to the root.

It is processed as the particular cases of the first search.

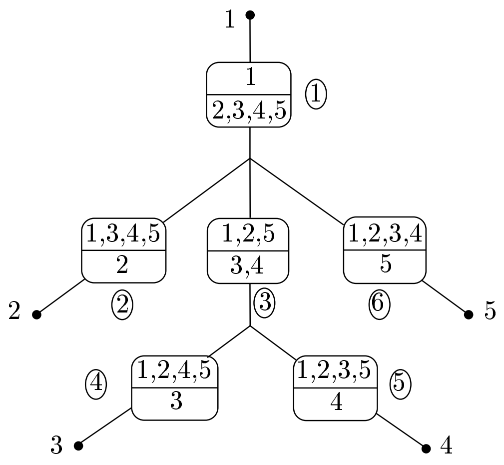


Figure 5: Preordered depth-first search (as shown by circled numbers). Information is passed for the lower subtrees from the root to the leaves.

The IQ^* algorithm uses the same two searches of T^*_{i-1} to obtain a suitable processing order for the components of the B^*_{i-1} bipartitions, each component being processed as indicated in Algorithm 1.

4.3 Complexity result

LEMMA 4.1 *The IQ^* algorithm examines each possible r4-tree once.*

Proof. Each r4-tree q is only considered at the step where the last species it references is considered. Let i be this step. Algorithm 1 insures that at this step, only the crossed quartets sets R^c_σ are examined. The recursive searches of T^*_{i-1} insure that each component σ, and thus each set R^c_σ is only considered once at step i. By definition, the r4-tree $q = xi|yz$ belongs to only one set $R^{j,k}_\sigma$ (the set $R^{j,k}_\sigma \,/\, x \in \sigma, y \in \overline{\sigma}_j, z \in \overline{\sigma}_k, j \neq k$), which in turn is included in only one set R^c_σ. □

THEOREM 4.1 *B^*, T^* and thus Q^* can be computed in time $O(n^4)$.*

Proof. Sets L_j are implemented as binary arrays, indicating the presence or absence of every species. At the beginning of each step considering a new species, these sets are emptied by a simple recursive search of the tree, which requires time $O(n^2)$ (proportionally to the summed sizes of the L_j's), thus $O(n^3)$ globally. Then, by Algorithm 1, some unions are performed between several of these sets. Each set L_j of a component σ_j is only added once to the sets L_σ of the components σ it contains. In the worst case, each component contains $O(n)$ other components, thus unions of L_j sets may require $O(n^3)$ by step, *i.e.*, $O(n^4)$ globally. Moreover, separate species x are also added in $O(1)$ to the L_j sets, each one possibly several times. In the worst case this happens each time an r4-tree is considered. From Lemma 4.1, each of the $3\binom{n}{4}$ r4-tree is considered only once during the whole algorithm, thus additions of species x to L_j sets requires $O(n^4)$ globally.

For each component σ_j of a given bipartition, we need to know the species X_j it contains, to enumerate easily the r4-trees of R^c_σ in Algorithm 1. Sets X_j are implemented as chained lists. Preliminary to each step, we initialize these sets by a simple recursive search of the tree, which requires $O(n^2)$, proportionally to the amount of information to be stored. Then generating the r4-trees of R^c_σ in the inner loop of Algorithm 1 is simply implemented by exploring the X_j sets, thus, by Lemma 4.1, this globally requires $O(n^4)$.

Again by Lemma 4.1, $O(n^4)$ requests to Q are performed. Each costs time $O(1)$ since we assume that the precise r4-tree (or the absence of r4-tree) inferred for each quartet is stored in a four-dimension array (if not, an $O(n^4)$ preprocessing insures this).

At the end of each step, T^*_i is obtained in time $O(n)$ from the answers collected for the various components and from T^*_{i-1}.

Thus, IQ^* is in time $O(n^4)$ and we can see from the above analysis that the leading coefficient is small (as confirmed by the running times). □

If need be, Q^* may be obtained in $O(|Q^*|)$ by performing two depth-first searches of T^* (as described in section 4.2) since Q^* exactly corresponds to the crossed r4-trees of T^*.

5 Reconstructing phylogenies by using the Q^* relation

5.1 Theoretical results on distance data

When applied to tree-like data, the Q^* relation enables us to recover the correct topology. However, for processing real data sets it is much more important that the algorithm be guaranteed to converge on the correct tree in the case of imperfect data. The convergence of a quartet-based method first depends on the convergence of the r4-tree inference method. We show below how we can derive a polynomial convergence rate for Q^* when inferring the r4-trees by a method (called FPM) based on a distance principle. For this purpose we use the recent results of Erdös *et al* [13] giving the probability under the Cavender-Farris model of evolution that FPM correctly infers an r4-tree.

Inferring the r4-trees from the data

Let $\hat{D}$ be a distance matrix obtained from character sequences available for the studied species. $\hat{D}$ is a *tree distance*, *i.e.*, defines a tree, iff it satisfies the four-point condition [7] (also called the additivity condition): for any four species x, y, z, t, the larger two of the three sums $\hat{D}_{xy}+\hat{D}_{zt}$, $\hat{D}_{xz}+\hat{D}_{yt}$, $\hat{D}_{xt}+\hat{D}_{yz}$ are equal. If $\hat{D}_{xy}+\hat{D}_{zt}$ is the smallest sum, then there must be at least one edge separating x, y from z, t in the tree representing $\hat{D}$. This topological constraint corresponds to the r4-tree $xy|zt$. In order to infer r4-trees from *any* (non-additive) distance matrix $\hat{D}$, the following empirical rule, called FPM, is commonly used [4, 7, 12, 13, 16, 27]:

$$\left(\hat{D}_{xy}+\hat{D}_{zt} < \left\{ \begin{array}{c} \hat{D}_{xz}+\hat{D}_{yt} \\ \hat{D}_{xt}+\hat{D}_{yz} \end{array} \right.\right) \Leftrightarrow xy|zt \ .$$

If none of the three sums is strictly lower than the others, then no r4-tree is inferred for the quartet. FPM is proven to be well-founded for various phylogenetic reconstruction criteria [17, 26].

The Cavender-Farris model of evolution

The *Cavender-Farris* model of evolution for binary sequences (*i.e.*, having 2-state characters) [9] associates independently with every edge e in a model tree T a probability $p(e)$ $(0 < p(e) < .5\ )$ of observing different states at its two end-points for any given site. The sites evolve identically and independently (i.i.d.) along each edge, and the states at the root of the tree have equal probability to be 0 or 1. The result of k sites evolving under this model is a set of binary sequences of length k at the leaves. Let $H(i,j)$ denote the *Hamming distance* between sequences i and j, let $L_{ij} = H(i,j)/k$ denote their *dissimilarity score*, and let p_{ij} denote the model probability of change between species i and j, so that $p_{ij} = \mathrm{E}(L_{ij})$. Interpreting the Cavender-Farris model of evolution as a Poissonian process leads to defining the *evolutive distance* between sequences i and j as $D_{ij} = -\frac{1}{2}\log(1-2p_{ij})$, which is the expected number of mutations between the two species. Similarly, the estimated evolutive distance between species i and j is obtained by $\hat{D}_{ij} = -\frac{1}{2}\log(1-2L_{ij})$.

The convergence rate of Q^*

Erdös *et al* used the Azuma-Hoeffding inequality (see [2]) to derive the probability that FPM gives the correct r4-tree for a given quartet:

THEOREM 5.1 (ERDÖS *et al* 97 (MODIFIED)) *Assume that for all edges e of a Cavender-Farris tree T the mutation probability $p(e)$ is bounded below by f, and let $d \geq max\ D_{ij}$ denote the maximum distance induced by the model* *between two species. The probability that FPM returns the appropriate subtree of T is at least* $1 - 18\exp(-(1-\sqrt{1-2f}\,)^2 e^{-4d} k/8)$.

The Q^* relation enables us to recover the entire topology of T iff all r4-trees of Q_T are correctly inferred. Thus, reasonably replacing $(1-\sqrt{1-2f}\,)$ by its first order approximation, f, we obtain

THEOREM 5.2 *Under the Cavender-Farris model of evolution, the probability that the Q^* relation recovers the entire topology of an unknown tree T is at least*

$$1 - 18\binom{n}{4}\exp(-f^2 e^{-4d} k/8)\ .$$

This enables us to establish the convergence rate of Q^*, *i.e.*, the number of characters needed to recover the entire topology of T:

THEOREM 5.3 *Suppose k sites evolve on a phylogeny T under the Cavender-Farris model, $T^* = T$ with probability at least $1-\epsilon$ $(\epsilon > 0)$ if*

$$k > \exp(4d)\,\frac{32\log(n) + 8\log(18/\epsilon)}{f^2}\,. \tag{1}$$

As often, the number of required characters grows logarithmically with the inverse of the desired confidence level.

Considering both d and f as constants when n increases is impossible. One may consider the problem from two points of view, depending on which of these parameters is fixed (if any).

In their convergence analysis [13], Erdös *et al* considered f as fixed and d as a parameter. They compared the convergence rate of their *Short Quartet* method to the one of the *Single Pivot* method [1], as used by Farach and Kannan [14]. This last method has the same convergence rate as Q^*, up to a constant [13]. From this standpoint, when d is not bounded, then the three methods have an exponential convergence rate. Assuming the edge lengths are bounded above gives a polynomial rate to the *Short Quartet* method since it only considers paths with a number of edges depending on the *depth* of the tree (*i.e.*, whose length is in $O(\log n)$). The two other methods have still an exponential rate since in the worst case $d \in O(n)$.

In practice, however, assuming that d is a small constant is very reasonable since biologists usually consider data sets with small $\hat{D}_{ij}$ values ($\hat{D}_{ij} < 1$, [24]) and are first confronted with problems resulting from the presence of small edges in the tree: increasing n necessarily leads to decreasing f (newly added species break existing edges). Thus, we reasonably consider d as a small constant and f as a parameter which can take arbitrarily small values. Choosing d as a constant implies that the three above mentioned methods have convergence rates of the same order. Moreover we know (within a first order) $f \leq O(1/\log n)$. In the best case $f = O(1/\log n)$ and the complexity is logarithmic However, this case is very unlikely, since it corresponds to a well balanced tree with edges of similar lengths. Basically, the sampling complexity of the methods is quadratic in $1/f$. For example, if we assume that $f = O(1/n)$, which may be seen as reasonable, the sampling complexity of the methods is in $O(n^2 \log n)$. However, f can be arbitrarily small, leading the above methods to require too high a number of characters to produce the correct tree with high probability. This problem is well-known to biologists and it is improbable that a method will exist that can produce a reliable estimate of phylogenies in this area of the parameter space.

Finally, note that Theorem 5.1 extends to general stochastic models [13], so the above results obtained for binary sequences generalize, *e.g.*, to 4-state or 20-state molecular data.

5.2 Experimental results

The condition for edges to be in T^* is that all r4-trees they induce be in the data set Q. This could seem too strict a constraint to produce any edge, and it is indeed the case for random data. For highly diverging sequences, *i.e.*, submitted to an high evolutive noise, it is likely that Q^* will only detect part of the inter-species relationships. However, most biological data sets do contain information that can be *extracted* by Q^*. We applied the IQ^* algorithm to several data sets taken from the literature and always obtained at least partly resolved trees. Some fully resolved trees were even inferred for high quality data sets. As an example, we present in Figure 6 the tree obtained on a mammal data set provided by D. Penny [25]. Most of the inferred inter-species relationships coïncide with what is expected, *e.g.*, primates (human, gorilla, ape) and ungulates (cow, sheep, pig, horse). The Dog-Kangaroo group, also inferred by other methods (NJ, parsimony, ...), may be suspected of being incorrect, but Penny *et al* [25] report that they did not reach any firm conclusion. The Q^* relation takes no decision concerning the position of the rodent and the rabbit, in relation to the other mammals. To date, their is no consensus on the position of the rodent and the recent polemic concerns the possibility of the rodent being at the root of the mammalian evolution [11, 18].

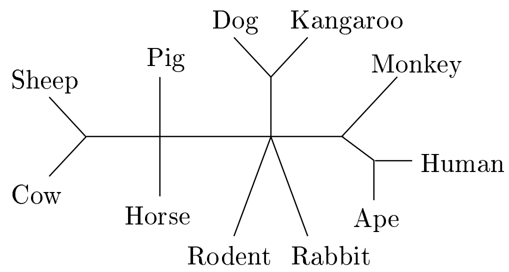


Figure 6: Applying Q^* to 11 mammal sequences

Running times of the method are reasonable. For example, a non-optimized version of IQ^* took 0.03 seconds (including the time consuming I/O operations) to compute the above tree on 11 species (330 r4-trees processed) on a SUN SPARC 5. In contrast, inference of the r4-trees, including distance corrections, took $0.07s$.

To test the accuracy of the method, we also performed simulations under various conditions of evolution according to [6, 20]. Results confirm that Q^* usually produces partially resolved trees, which possess almost only safe branches. This contrasts with usual reconstruction methods which usually propose several wrong edges in a tree. Therefore, even if one aims at a more resolved tree than the one provided by the Q^* method, one may still consider the tree T^* it proposes as a safe basis for adding new edges.

Acknowledgment

The authors would like to thank H.-J. Bandelt for preliminary discussion, Tandy Warnow for kindly sending us a draft of the paper [13], and D. Penny for providing the mammal data set.

References

[1] R. Agarwala, V. Bafna, M. Farach, B. Narayanan, M. Paterson, and M. Thorup. On the approximability of numerical taxonomy: Fitting distances by tree metrics. In *Proc. of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1996.

[2] N. Alon and J.H. Spencer. *The Probabilistic Method.* John Wiley and Sons, New York, 1992.

[3] K. Atteson. Workshop on mathematical hierarchies and biology. DIMACS, november 1996.

[4] H-J. Bandelt and A. Dress. Reconstructing the shape of a tree from observed dissimilarity data. *Adv. in appl. math.*, 7:309–343, 1986.

[5] H.-J. Bandelt and A. Dress. A canonical decomposition theory for metrics on a finite set. *Advances Math*, 92:47–105, 1992.

[6] V. Berry and O. Gascuel. On the interpretation of bootstrap trees: appropriate threshold of clade selection and induced gain. *Mol. Biol. Evol.*, 13(7):999–1011, 1996.

[7] P. Buneman. *Mathematics in Archeological and Historical Sciences*, chapter The recovery of trees from measures of dissimilarity, pages 387–395. Edhinburgh University Press, 1971.

[8] P. Buneman. A note on metric properties of trees. *J. of Comb. Theory*, 17(B):48–50, 1974.

[9] J. A. Cavender. Taxonomy with confidence. *Math. Biosci.*, 40:271–280, 1978.

[10] H. Colonius and H.H. Schulze. Tree structures for proximity data. *British Journal of Math. and Stat. Psychology*, 34:167–180, 1981.

[11] A.M. d'Erchia, C. Gissi, G. Pesole, C. Saccone, and U. Arnason. The guinea-pig is not a rodent. *Nature*, 381:597–600, 1996.

[12] A. Dress, A. von Haeseler, and M. Krueger. Reconstructing phylogenetic trees using variants of the "four-point condition". *Studien zur Klassifikation*, pages 299–305, 1986.

[13] P.L. Erdös, M.A. Steel, L.A. Székely, and T.J. Warnow. Constructing big trees from short sequences. Technical report, DIMACS, 1997.

[14] M. Farach and S. Kannan. Efficient algorithms for inverting evolution. In *Proceedings of the ACM Symposium on the Foundations of Computer Science*, pages 230–236, 1996.

[15] J. Felsenstein. Phylogenies from molecular sequences: inference and reliability. *Annu. Rev. Genet.*, 22:521–565, 1988.

[16] W. M. Fitch. A non-sequential method for constructing trees and hierarchical classifications. *J. Mol. Evol.*, 18:30–37, 1981.

[17] O. Gascuel and D. Levy. A reduction for approximating a (non-metric) dissimilarity by a tree distance. *J. of Classification*, 1996.

[18] D. Graur, W.A. Hide, and W.-H. Li. Is the guinea-pig a rodent? *Nature*, 351:649–652, 1991.

[19] D. Gusfield. Efficient algorithms for inferring evolutionnary trees. *Networks*, 21:19–28, 1991.

[20] M.K. Kuhner and J. Felsenstein. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol. Biol. Evol.*, 11:459–468, 1994.

[21] G. Lecointre, H. Philippe, H.L.V. Lê, and H. Le Guyader. Species sampling has a major impact on phylogenetic inference. *Mol. Phylogenet. Evol.*, 2:205–224, 1993.

[22] C. Meacham. A manual method for character compatibility. *Taxon*, 30:591–600, 1981.

[23] V. Moulton and M.A. Steel. Retractions of finite distance functions onto tree metrics. Technical Report 148, Univ. of Canterbury, 1996.

[24] M. Nei. Relative efficiencies of different tree-making methods for molecular data. In M.M. Miyamoto and J. Cracraft, editors, *Phylogenetic analysis of DNA sequences*. Oxford Univ. Press, 1991.

[25] D. Penny, M.D. Hendy, and M.A. Steel. Testing the theory of descent. In M. M. Miyamoto and J. Cracraft, editors, *Phylogenetic analysis of DNA sequences*, pages 155–183. 1991.

[26] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstruction phylogenetic trees. *Mol. Biol. Evol.*, 4(4):406–425, 1987.

[27] S. Sattah and A. Tversky. Additive similarity trees. *Psychometrika*, 42:319–345, 1977.

[28] M. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *J. of Classification*, 9:91–116, 1992.

[29] K. Strimmer and A. von Haeseler. Quartet puzzling: a quartet maximum-likelihood method for reconstructing tree topologies. *Mol. Biol. Evol.*, 13(7):964–969, 1996.

[30] David L. Swofford and Gary J. Olsen. Phylogeny reconstruction. In David M. Hillis and Craig Moritz, editors, *Molecular Systematics*, chapter 11, pages 411–501. 1990.