Inferring Evolutionary Trees with Strong Combinatorial Evidence

Vincent Berry* Olivier Gascuel†

Abstract

We consider the problem of inferring the evolutionary tree of a set of n species. We propose a quartet reconstruction method which specifically produces trees whose edges have strong combinatorial evidence. Let Q be a set of resolved quartets defined on the studied species, the method computes the unique maximum subset Q^* of Q which is equivalent to a tree and outputs the corresponding tree as an estimate of the species' phylogeny. We use a characterization of the subset Q^* due to [6] to provide an $O(n^4)$ incremental algorithm for this variant of the NP-hard quartet consistency problem. Moreover, when chosing the resolution of the quartets by the *Four-Point Method* (FPM) and considering the Cavender-Farris model of evolution, we show that the convergence rate of the Q^* *method* is at worst polynomial when the maximum evolutive distance between two species is bounded. We complete these theoretical results by an experimental study on real and simulated data sets. The results show that *i)* as expected, the strong combinatorial constraints it imposes on each edge leads the Q^* method to propose very few incorrect edges; *ii)* more surprisingly, the method infers trees with a relatively high degree of resolution.

Keywords: phylogeny reconstruction, quartet method, exact polynomial algorithm, partially resolved tree, combinatorial technique, worst-case convergence rate, experimental study.

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 consist of

[*] Corr. author, Dept. of Computer Science University of Warwick, UK, vincent@dcs.warwick.ac.uk. This work was partly supported by ESPRIT LTR Project no. 20244 — ALCOM-IT.

[†] Département d'Informatique Fondamentale, LIRMM, Université de Montpellier II, France.

homologous DNA sequences taken 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. An excellent overview on phylogenetic reconstruction criteria and algorithms can be found in [48].

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 by relying on a classical result from dissimilarity analysis [8, 17]. Farach and Kannan [23] proved the first guaranteed convergence rate (for the algorithm of [1] that they called *Single Pivot*, or *SP*) by introducing a variational distance between trees. More recently, Ambainis *et al* [4] improved this result by showing that if the solution provided by *SP* is improved to its local optimum, then the convergence rate of the method is within a constant factor of the best achievable rate, provided that the species' phylogeny does not contain very short edges. Erdös *et al* [22] produced a quartet method (the *Short Quartet Method*, or *SQM*) and gave a bound on its convergence rate in the sense of the L_∞ metric, for the problem of recovering the topology of the species' phylogeny. The bound they obtain is, in some cases, better than the one derived for the *SP* method for this problem. The improvement results from the fact that the bound on the convergence rate of *SQM* depends on the depth (the rank) of the phylogeny, while the bound for the *SP* method depends on the diameter. For the same problem, Atteson [5] proved a bound on the convergence rate for two of the most popular distance-based algorithms used by practitioners, namely *Neighbor Joining* [44] and *Addtree* [45], while Kearney provided a bound on the convergence rate of methods utilizing ordinal assertions [35]. These sampling complexity results are of importance in the phylogenetic domain since the amount of available data (*i.e.*, the sequence length) is a very critical resource.

We investigate here a phylogenetic reconstruction method which specifically produces trees whose edges have a strong combinatorial support. The method is based on a quaternary relation introduced by Bandelt and Dress [6], and thus relies on a quartet reconstruction principle [6, 18, 22, 45, 46, 47].

Quartet methods first compute subtrees of the phylogeny which correspond to subsets of 4 species. Then they rely on a combinatorial algorithm to construct a phylogeny on the entire set of species, respecting as many as possible of the (possibly conflicting) structural constraints imposed by the subtrees on four species. These subtrees can be of two kinds: *resolved* (*i.e.*, having an edge separating two species from the two others, cf. Fig. 1 (a-c)), which corresponds to the assumption that two species belong to a distinct group of species (*e.g.*, carnivores) than the two others; or *unresolved* (*i.e.*, having no edge separating two species from the others, cf. Fig. 1 (d)). Subtrees of the second kind only express uncertainty with respect to which species belong to the same group, and are not considered as structural constraints. Indeed, the phylogeny of a set of species is usually assumed to be binary, so that for any four species, two of them belong to a group excluding the two others. Thus, only resolved subtrees on four species (that we call *r4-trees*) are considered to be of importance.

The interest in quartet-based methods is that many methods exist which can efficiently infer trees 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 r4-trees they infer.

A phylogeny is said to be *consistent* with (or to *induce*) an r4-tree if at least one of its edges separates the concerned four species in the same way as the r4-tree. Any phylogeny is uniquely defined by the set of r4-trees it induces [18]. Such an r4-tree set, *i.e.*, which corresponds exactly with a phylogeny on the entire set of species, is said to be *tree-like*. Knowing whether an r4-tree set Q is tree-like is polynomial, as well as reconstructing the tree to which it is equivalent [6]. Less restrictively, given an r4-tree set Q, knowing if there exists a phylogeny consistent with its r4-trees (but possibly inducing more r4-trees) is an NP-complete problem [46]. Several polynomial-time heuristics have been designed to solve the corresponding NP-hard optimization problem, *i.e.*, finding a maximum subset of Q which is consistent with a phylogeny, or some of its variations [6, 22, 47]. Here we consider the problem of finding the maximum subset of Q which is tree-like [6, 8, 14]. We call this subset Q^*.

The interest in the subset Q^* for phylogeny reconstruction relies on several points: first, it corresponds to the maximum tree-like part that we can extract from the data, *i.e.* to the greatest part of the data which fully corresponds to the model we chose for representing the species' history. Moreover, the tree to which Q^* corresponds, called T^*, has the interesting property that it does not contradict any piece of data (any r4-tree of the input set Q), nor does it necessitate any new hypothesis (it does not induce any r4-tree not present in Q). Lastly, as long as the method inferring the r4-trees from the biological data is not biased, T^* can be seen as a safe estimate of the species' phylogeny, due to the stringent combinatorial constraints imposed on its internal edges, *i.e.*, each one must respect between $O(n^2)$ and $\Omega(n^4)$ data r4-trees. In this way, few wrong edges are likely to be inferred due to random sampling errors. However, since T^* contains only edges showing strong convincing evidence, the chances are that this tree will be poorly resolved and will only recover a small part of the species' phylogeny. In practice, T^* is actually partially resolved, but it nevertheless contains a reasonably high number of edges (see last section), so that it is likely to recover a non-negligible part of the estimated phylogeny.

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

Until now, no algorithm has been given in the various papers mentioning the problem [6, 8, 14], though this problem was thought to be polynomial (Bandelt and Steel, personal communications). In this paper, we provide an $O(n^4)$ *incremental* algorithm, called IQ^*, to compute T^*: starting from four species, T^* is built up by progressively attaching the remaining species to the tree. In most usual cases, $|Q| \in O(n^4)$, and the IQ^* algorithm then has an optimal complexity bound. 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 thus obtain for computing the maximum tree-like subset of Q, is the same as that required by the best algorithm [6] (to our knowledge) which enables to decide whether an r4-tree set Q is tree-like.

Moreover, using Hoeffding's inequality, we give a bound on the convergence rate of the IQ^* algorithm when associated with a distance-based r4-tree inference method under the Cavender-Farris model of evolution, in a similar way as [5, 22]. This bound is polynomial when the evolutive distances between the studied species are bounded, as is usually the case in practice [42].

In the following, we first give prerequisites (section 2) and introduce the incremental principle enabling us to compute Q^* in $O(n^5)$ (section 3). Next, we improve this result by giving the $O(n^4)$ algorithm (section 4). Lastly, we concentrate on the use of Q^* for phylogeny reconstruction, giving a bound on its convergence rate (section 5) and reporting experimental results from real and simulated data (section 6).

2 Preliminaries

In this section, we present the basics of reconstructing phylogenies from r4-trees.

Definition 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 (Fig. 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 $xy|zt \equiv yx|zt \equiv zt|yx$). These topologies are called *r4-trees* (for *resolved 4-trees*). Let Q be a set of r4-trees defined on S. Q can be seen as a set of topological constraints to respect when constructing the tree on the entire set S of species. The set Q is said to be *complete*, when it contains an r4-tree for each quartet of species. When Q is not complete, we consider the *unresolved* quartets as associated with the unresolved *star* topology (with one internal node of degree 4). Other approaches could be investigated, but they are close (or identical) to the quartet tree-consistency problem and, therefore, seem difficult to deal with from a computational standpoint.

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 degree ≥ 4) of T corresponds to the star topology, thus T induces no r4-tree for the corresponding quartet. Let Q_T denote the set of r4-trees induced by T, Q_T contains at most one r4-tree for each quartet of species. Similarly, we will only consider r4-tree sets Q containing at most one r4-tree for each quartet of species.

A phylogeny T can also be characterized by the set of bipartitions its edges induce on the set S of species [6, 14]. Indeed, deleting any edge from T disconnects T into two components, and thereby induces a bipartition on the whole set S. Its two parts correspond respectively to the species of the two components. A bipartition is called *trivial* when one of its components contains less than 2 species, and hence induces no r4-tree. To each bipartition $b = \sigma|\overline{\sigma}$ we associate the set $Q_b = \{xy|zt \text{ s.t. } x, y \in \sigma,\ z, t \in \overline{\sigma}\}$ of r4-trees it *induces*. For a set B of bipartitions, we define $Q_B = \cup_{b \in B} Q_b$. 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.

A set of bipartitions is *tree-like* (or *tree-compatible*) iff there exists a tree with edges corresponding to these bipartitions. The following are well-known results that we will need further on:

Lemma 1 ([14])

- *Two bipartitions* $b_1 : \sigma_1|\overline{\sigma}_1, b_2 : \sigma_2|\overline{\sigma}_2$ *are tree-compatible iff at least one of* $\sigma_1 \cap \sigma_2$, $\sigma_1 \cap \overline{\sigma}_2$, $\overline{\sigma}_1 \cap \sigma_2$, $\overline{\sigma}_1 \cap \overline{\sigma}_2$ *is empty.*
- *A set B of bipartitions is tree-compatible iff every pair of bipartitions* $b_1, b_2 \in B$ *are tree-compatible (*i.e.*, we only need to check the compatibility of subsets of two elements to decide for the compatibility of the whole set).*

Corollary 1 *A set B of bipartitions on a given set of species is tree-compatible iff Q_B contains at most one r4-tree for each quartet of species.*

Definition 2 *An r4-tree set Q is*

- tree-consistent *iff there exists a tree T such that* $Q \subseteq Q_T$
- tree-like *iff there exists a tree T such that* $Q_T = Q$.

These two notions are computationally very different since knowing if a given r4-tree set Q is tree-like is polynomial [6, 18] (as for a bipartition set [15]), whereas knowing if Q is tree-consistent is NP-complete in general (except, *e.g.*, when Q is complete) [46].

Finding the maximum tree-consistent subset of an r4-tree set Q is thus NP-hard. Here we consider the problem of finding *the maximum tree-like subset of* Q, that we note Q^*. The uniqueness of this set is surprising but derives from the characterization Bandelt and Dress [6] gave for Q^*. They originally defined Q^* as:

Definition 3 ([6]) *Let Q be an r4-tree set and B^* be the set of bipartitions $b = \sigma|\overline{\sigma}$ such that $Q_b \subseteq Q$, then $Q^* = \bigcup_{b \in B^*} Q_b$.*

Corollary 2 *Q^* is the maximum subset of Q which is tree-like,* i.e., *there exists some tree T^* with $Q_{T^*} = Q^* \subseteq Q$ and $\forall Q' \subseteq Q$, if Q' is tree-like then $Q' \subseteq Q^*$ (and $|Q'| \leq |Q^*|$).*

This result (including the uniqueness of Q^*) derives from the fact that B^* is a set of tree-compatible bipartitions (due to corollary 1 and to the fact that Q contains at most one r4-tree for each quartet of species). Moreover, the tree-like subsets of Q correspond bijectively with the subsets of B^*. Thus, tree-like subsets of Q form a lattice having as unique maximal (and maximum) element the set Q^* corresponding to the complete set B^*.

Obtaining T^*, Q^* or B^* from one another is easy, *e.g.*, B^* gives T^* in linear time [30, 40] and T^* gives Q^* in linear time as well (see section 4). Trivially, when Q is tree-like, $Q^* = Q$. 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. Note also that Q^* is never the *maximum* tree-consistent subset of Q (except when $Q^* = Q$), since it is not even a *maximal* tree-consistent subset of Q: $\forall r \in Q - Q^*$, $\{r\} \cup Q^*$ is tree-consistent. Moreover, we can easily find counter-examples showing that Q^* is not always contained in the maximum set of tree-consistent r4-trees.

The subset Q^* is also related to the work of Buneman [14], who proposed a way to infer a set of tree-compatible bipartitions from a dissimilarity matrix d. This set of bipartitions is defined in such a way as to correspond to the set B^* when the r4-trees of Q are inferred from d by a simple distance principle (that we detail in section 5).

3 Computing Q^* by an incremental principle

In a context sharing similarities with that of Q^*, Bandelt and Dress mention an $O(n^6)$ incremental algorithm to obtain the $O(n^2)$ *d-splits* of a distance matrix d [7]. When applying the same principle to the r4-tree set Q, we derive a polynomial algorithm to obtain the $O(n)$ bipartitions of B^* Note that Bandelt also thought this to be possible but did not publish his result (Bandelt, personal communication). We first show that this incremental approach is correct for computing Q^*. Then, we give the simple $O(n^5)$ algorithm which results. In the next section, we will show how a more sophisticated $O(n^4)$ algorithm can be obtained.

Definition 4 *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 of* Q *only* referencing *species in* $[i]$, *i.e.,* $\forall\, xy|zt \in Q_{[i]},\ \{x, y, z, t\} \subseteq [i]$.
- Q_i *denotes the subset of r4-trees of* $Q_{[i]}$ *referencing species* i, *i.e.,* $\forall\, xy|zt \in Q_i,\ i \in \{x, y, z, t\}$.
- $B^*_{[i]}$ *is the set of bipartitions* $b = \sigma|\overline{\sigma}$ *on* $[i]$ *s.t.* $Q_b \subseteq Q_{[i]}$. *Note that* $B^*_{[i]}$ *contains at least the* i *trivial bipartitions* $\{1\}|\{2, \ldots, i\}, \ldots, \{i\}|\{1, \ldots, i-1\}$.
- $Q^*_{[i]} = \bigcup_{b \in B^*_{[i]}} Q_b$.

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 $Q^*_{[n]} = Q^*$.

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 (say 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 5 *Let* b *be a bipartition on* $[i]$, $b_{\setminus i}$ *denotes its* restriction *to the species* $[i–1]$. *In a wider sense,* $B^*_{\setminus i}$ *denotes the bipartitions of* $B^*_{[i]}$ *restricted to* $[i-1]$.

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

Lemma 2 $B^*_{\setminus i} \subseteq B^*_{[i-1]}$.

PROOF. Let $b = \sigma|\overline{\sigma}$ be a bipartition of $B^*_{[i]}$. If b is trivial then $b_{\setminus i}$ 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. $b_{\setminus i} = \sigma - \{i\}|\overline{\sigma}$ is trivial ($|\sigma - \{i\}| = 1$) implying $b_{\setminus i} \in B^*_{[i-1]}$, or

2. $b_{\setminus i}$ is not trivial and $\forall x, y \in \sigma - \{i\}$ and $z, t \in \overline{\sigma}$, we have $\{x, y, z, t\} \subseteq [i-1]$ and $xy|zt \in Q_{[i]}$, thus more precisely $xy|zt \in Q_{[i-1]}$. Then by definition, $b_{\setminus i} \in B^*_{[i-1]}$. □

The lemma cannot be extended to state that $B^*_{\setminus i} = B^*_{[i-1]}$, as the following example shows: let $Q = \{12|34, 15|23, 15|24, 15|34\}$, then $B^*_{[4]}$ contains $\{12\}|\{34\}$ plus trivial bipartitions on [4], $B^*_{[5]}$ contains $\{15\}|\{234\}$ plus trivial bipartitions on [5], while $B^*_{\setminus 5}$ only contains trivial bipartitions on [4]. We thus have $b = \{1, 2\}|\{3, 4\} \in B^*_{[4]}$ but $b \notin B^*_{\setminus 5}$ [1], showing that in some cases $B^*_{\setminus i} \neq B^*_{[i-1]}$.

Corollary 3 *Any bipartition of $B^*_{[i]}$ can be directly obtained from $B^*_{[i-1]}$, i.e., $\forall b \in B^*_{[i]}$, $\exists\, b' = \sigma|\overline{\sigma} \in B^*_{[i-1]}$ s.t. $b = \sigma \cup \{i\}|\overline{\sigma}$ 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. $b = \sigma|\overline{\sigma} \cup \{i\}$) can 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 first incremental algorithm

The algorithm first considers the trivial case of four species, where $B^*_{[4]}$ is readily obtained: it contains the five trivial bipartitions on these species plus possibly the bipartition $\{x, y\}|\{z, t\}$, with $\{x, y, z, t\} = \{1, 2, 3, 4\}$ (if $xy|zt \in Q$).

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, $B^*_{[i]}$ is initialized with the trivial bipartition $\{i\}|\{1, \ldots, i-1\}$, then for each bipartition $b = \sigma|\overline{\sigma} \in B^*_{[i-1]}$, we 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.

[1] to have $b \in B^*_{\setminus 5}$ would require that $\{125\}|\{34\} \in B^*_{[5]}$ (impossible since $\{25|34, 12|35\} \notin Q$) or $\{12\}|\{345\} \in B^*_{[5]}$ (impossible since $\{12|35, 12|45\} \notin Q$)

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 thus 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 $O(n^4)$ incremental algorithm

We now present the IQ^* algorithm which consists of 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, *wrt* the previous algorithm, is to focus simultaneously on $T^*_{[i]}$ and $B^*_{[i]}$. We can then guarantee that each r4-tree is only processed once, by searching the tree $T^*_{[i]}$ in a specific order and sharing information between the *components* of $B^*_{[i-1]}$ bipartitions, on the basis of the r4-trees they commonly concern.

4.1 Basic principles

Definition 6 *Let i be the species currently considered and $b = \sigma|\overline{\sigma}$ a bipartition on the species $[i-1]$;*

- *σ and $\overline{\sigma}$ are called the parts or the* components *of bipartition b.*
- *we say that the r4-trees $xi|yz$ are* required *(to be in Q) by the* component *σ of b if they are induced by the addition of i to this component, i.e., if $x \in \sigma$ and $y, z \in \overline{\sigma}$. We let R_σ denote the set of r4-trees required by σ. Note that this definition only considers newly required r4-trees (i.e., referencing the species i) since the r4-trees on $[i-1]$ need not be reexamined as stated in the previous section.*
- *r4-trees are called* processed *when their presence in Q has been examined.*
- *In the following, the term* subtrees *only denotes the connected component of a tree T that can be obtained by deleting one of its edges.*

Each r4-tree may be required by components of several bipartitions. For example, in Fig. 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 $xi|yz$.

How do we know all the components that require a given r4-tree? Simply, by taking advantage of the containment relations existing between components. For example, in Fig. 2, $x \in \sigma_1 \subset \sigma_2$, 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 $B^*_{[i-1]}$ 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. In the following, we will sometimes confuse the components with their associated subtrees and use one term or the other depending on the context. Thus we only have to follow the $T^*_{[i-1]}$ topology to obtain the containment relations between components from the ones induced between subtrees. For example, in Fig. 2 every 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.

Figure 3 illustrates the situation arising around any given internal node r, connected to edges $e_\sigma, e_1 \dots e_l$. To these edges correspond the components $\sigma, \overline{\sigma}, \sigma_1, \overline{\sigma}_1, \dots, \sigma_l, \overline{\sigma}_l$. The r4-trees $xi|yz$ required by any $\sigma_j, j \in 1..l$ and such that $x \in \sigma$, are also required by σ. We then easily see the interest that lies in an ordered processing of the $B^*_{[i-1]}$ components to obtain $B^*_{[i]}$: suppose that the components $\sigma_1, \dots, \sigma_l$ are processed before σ, the knowledge of whether the species i can be added to $\sigma_1, \dots, \sigma_l$ can 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 the components containing σ should be processed before σ, to avoid at most as possible questioning Q when σ is considered. This naturally calls 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 of 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 $\nexists\, \theta$ s.t. $\sigma_j \supset \theta \supset \sigma$), and their *opposite* components, $\overline{\sigma}_j$, *minimally* contained in $\overline{\sigma}$ ($\overline{\sigma}_j \subset \overline{\sigma}$ and $\nexists\, \theta$ s.t. $\overline{\sigma}_j \subset \theta \subset \overline{\sigma}$).

Definition 7 *Let σ be a component contained in components $\sigma_1, \dots, \sigma_l$, we call* crossed *r4-trees of σ the r4-trees of the set $R^c_\sigma = \bigcup_{j,k} R^{j,k}_\sigma$ ($j \neq k \in 1..l$), where $R^{j,k}_\sigma = \{xi|yz$ s.t. $x \in \sigma$, $y \in \overline{\sigma}_j$, $z \in \overline{\sigma}_k\}$.*

From definitions 6 and 7 and from the simple containment relations existing between components we can directly derive:

Claim 1

- $R^c_\sigma \subseteq R_\sigma$.
- *If components* $\sigma_1, \ldots, \sigma_l$ *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 contain 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)* When 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, previously examined at step i. It now remains to be specified: *a)* How we obtain a processing order of the components ensuring that each component be considered after the components which contains it; *b)* How we pass information between components, to use the result of the previous processing of r4-trees they commonly require.

4.2 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 to σ. More precisely, r4-trees $xi|yz \in R_{\sigma_j}$ such that $x \in \sigma_j - \sigma$ are not required by σ. As a consequence, when σ_j cannot be extended, we cannot systematically deduce the same answer for σ, since the negative answer we got for σ_j might just result from r4-trees not required by σ. Let $L_j = \{x$ s.t. $\exists\, xi|yz \in R_{\sigma_j} - Q\}$ 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 cannot be extended. But only when $L_j \cap \sigma \neq \emptyset$ can we deduce from this result that σ cannot 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 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 paradoxical situation may arise where all components containing σ can be extended and yet σ cannot, 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 Fig. 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 cannot be extended to include i, but $\sigma \subset \sigma_j$ can. When considering this situation from the subtree standpoint, this again seems paradoxical. However, as remarked in section 3, some bipartitions of $B^*_{[i-1]}$ 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 $\sigma_1, \ldots, \sigma_l$ which contain it have already been processed.

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

4.3 Exploring the components in a specific order

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 common bipartition cannot 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.

Choosing an arbitrary leaf of the tree as a root enables us to direct the tree. To each edge of the tree are now associated a *lower* subtree and an *upper* subtree (the part of the tree opposite to the former with respect to the edge). The first recursive search of the tree, implemented as a *postordered depth-first search*, computes the information associated to all upper subtrees, using for each one the information already obtained for its *children* subtrees. For example, 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 (Fig. 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). For example, 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 particular cases of the first search, *i.e.*, we know it contains all species except the one at the root (*e.g.*, in Fig. 5, this subtree is labelled $\{2,3,4,5\}$).

4.4 The IQ^* algorithm

We now give the $O(n^4)$ IQ^* algorithm resulting from the previous sections. The main difference with the $O(n^5)$ algorithm of section 3.2 lies in the way the components of the $B^*_{[i-1]}$ bipartitions are processed. At each step i, the IQ^* algorithm explores the tree $T^*_{[i-1]}$ using the two postordered and preordered searches described above. It thus obtains a suitable processing order for the components which ensures that information can be shared from one component to another. As a result, each r4-tree is only examined once. Algorithm 2 summarizes the principle of IQ^*.

In Algorithm 3, we detail how IQ^* performs the postordered search of the tree $T^*_{[i-1]}$, using the notation of Fig. 3, for which we now assume that component σ contains the root of the tree. We do not detail the preordered search since it is symmetric with the postordered one, except that the component σ is processed before calling the recursive search. Moreover, edges e_j, e_k at step 6 are no longer child edges of the processed edge, but its sister or father edges in the tree. For example, in Fig. 3, if $PreOrder(e_1)$ is performed, $\{j,k\}$ are taken in $\{\sigma, 2 \dots l\}$.

4.5 Complexity result

We show here that the complexity of the IQ^* algorithm is in $O(n^4)$. This result mainly relies on the fact that each of the $3\binom{n}{4}$ possible r4-trees is considered only once, as stated below.

Lemma 3 *The IQ^* algorithm examines each possible r4-tree once.*

Proof. Each possible r4-tree $r = xi|yz$ is only considered at the step (let i) where the species of highest rank it references is added to the tree (*i.e.*, $x,y,z < i$). At this step, only crossed r4-trees are examined during the postordered search of $T^*_{[i-1]}$ and the symmetric preordered search (line 7 of Algorithm 3). These two searches ensure that each component σ is considered once and thus, from algorithm 3, each set $R^{j,k}_{\sigma}$ of crossed r4-trees is also considered once. Since, by definition, $r = xi|yz$ belongs to only one of these sets (the set $R_\sigma^{j,k}$ s.t. $x \in \sigma, y \in \overline{\sigma}_j, z \in \overline{\sigma}_k, j \neq k$), r is considered once during the whole algorithm.

Theorem 1 *B^*, T^* and thus Q^* can be computed in time $O(n^4)$.*

The detailed proof can be found in the appendix. Moreover, Q^* can be obtained in $O(|Q^*|)$ from T^* by simply listing its crossed r4-trees, using the same two depth-first searches as described in the previous section.

5 Convergence rate of IQ^* with *FPM* for sequence data

Here we consider applying the IQ^* algorithm to the problem of recovering the unknown phylogeny of a set of species. Phylogenies are usually reconstructed from sequences of molecular characters, taken from the DNA of the studied species. These sequences are transformed into evolutionary distances between species by using a given model of evolution. Then, r4-trees can be inferred on the basis of these distances. We chose here the *Four Point Method* (noted *FPM* and also known as the *weak Four-Point condition Method*) to infer the r4-trees. We call *Q^* method* the phylogenetic reconstruction method which results from applying *FPM*, then IQ^*.

An important feature for a phylogenetic reconstruction method is to be consistent, *i.e.* to converge on the correct phylogeny when more and more data are available. Here, we are particularly interested in the consistency of the method to reconstruct the structure (*i.e.*, the edges) of the phylogeny. As we have only a limited number of characters at disposal, it is also important that the method converges fastly. We show in this section how to derive a bound on the convergence rate of the Q^* method under the Cavender-Farris model of evolution. For this purpose, we use proof techniques similar to those investigated for other phylogenetic reconstruction methods [5, 22]. The following definitions are needed for the results of this section:

Definition 8 *Let T be a phylogeny on a set S of species,*

- *we note E_T the set of its internal edges and $l(e) \geq 0$ the length of an edge $e \in E_T$;*
- *D^T denotes the tree distance between species induced by the phylogeny T through the path-length metric, i.e., if we note $[xy]$ the path between species x and y in T, $D^T_{xy} = \sum_{e \in [xy]} l(e)$;*
- *Let D and D' be two distances (or dissimilarities) on the species S, $L_\infty(D, D') = \max_{x,y \in S} |D_{xy} - D'_{xy}|$.*

5.1 The Cavender-Farris model of evolution

The model of evolution known as the *Cavender-Farris model* [16, 24] is concerned with sequences of *binary* characters (having only state 0 or 1) and is a simplification of a previous model defined by Jukes and Cantor [34] for sequences of *four*-state characters (A,C,G,T). The reason for resorting to the two-state Cavender-Farris model is that the four basic molecular states can be partitionned into two groups: purines (A and G) and pyrimidines (C and T). Substitutions between states of the same group, numerous and of poor information, are sometimes ignored by biologists who then only consider substitutions from one group to the other.

The *Cavender-Farris* model of evolution assumes that a sequence of characters evolves from the root to the leaves of a model tree T, the characters evolving identically and independently (i.i.d.) along its edges and the states at its root having equal probability to be 0 or 1. The model associates independently with every edge e in T a probability $p(e)$ $(0<p(e)<.5)$ of observing different states at its two end-points for any given site of the sequence. This probability is lower than the expected number of substitutions which effectively occurs along e on this site, since multiple changes can lead to observe a similar state for the site at the two end-points of e. Let $l(e)$ be the expected number of substitutions along edge e, we have the formula $l(e) = -\frac{1}{2}\ln(1-2p(e))$. This formula is obtained by considering the Cavender-Farris model of evolution as a Poisson process. The tree T, associated with valuations $l(e)$, defines the tree distance D^T (cf. definition 8) that we seek to retrieve and which can be estimated from the sequences.

The result of k characters evolving under the Cavender-Farris model is a set of binary sequences of length k obtained at the leaves of T. Let f_{xy} be the average frequency with which we observe a difference for a site between sequences of the species x and y, and let p_{xy} denote the model probability of observing a change between x and y, so that $p_{xy} = \mathrm{E}(f_{xy})$. The *evolutive distance* D^T_{xy} between the two species, *i.e.*, the expected number of mutations between x and y, is given by $D^T_{xy} = -\frac{1}{2}\ln(1-2p_{xy})$. Therefore, the *estimated* evolutive distance between species x and y is obtained by $\hat{D}_{xy} = -\frac{1}{2}\ln(1-2f_{xy})$.

5.2 Inferring the r4-trees through the *Four Point Method*

Let $\hat{D}$ be a distance (or dissimilarity) matrix obtained from character sequences available for the studied species. $\hat{D}$ is a tree distance, *i.e.*, is represented by a unique positively valued tree, iff it satisfies the four-point condition [14] (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 $xy|zt$.

The r4-tree inference method *FPM* [6, 14, 22, 45] is designed on this basis, but can be applied to any dissimilarity matrix $\hat{D}$ and not only to tree distances:

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

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. For example, it is shown that *FPM* systematically indicates the same r4-tree as the least-square criterion (with the positivity constraints) [28] and the minimum evolution criterion [44].

5.3 Convergence results

We proceed in several steps: first, we give the condition under which *FPM* correctly recovers an r4-tree as a function of the evolutive distance between studied species; then, we give a bound on the probability that any of these distances are accurately estimated; last, we obtain the probability that the species' phylogeny is fully (or partially) recovered as a function of the number of characters, which equivalently gives us the number of characters required to correctly infer the phylogeny with a fixed probability.

Lemma 4 (Erdös *et al* 97, modified) *Let T be a phylogeny on species S, let $\hat{D}$ be an estimate of D^T and let $x,y,z,t \in S$. If $xy|zt \in Q_T$ and $L_\infty(D^T,\hat{D}) < \sum_{e\in[xz]\cap[yt]} \frac{l(e)}{2}$, then* FPM *returns the correct r4-tree $xy|zt$ for the quartet x,y,z,t.*

The following result can be shown (see appendix) by using Hoeffding's third inequality [32]:

Lemma 5 *Let T be a Cavender-Farris tree, and let $d = max\, D^T_{xy}$ denote the maximum distance induced by the model between two species. Let $\hat{D}$ be the estimated evolutive distance observed between species S on the basis of k characters and corrected according to the Cavender-Farris model. For two species x and y, we have*

$$P(|D^T_{xy} - \hat{D}_{xy}| < \epsilon) \geq 1 - 2\, e^{-(e^{-2\epsilon}-1)^2 e^{-4d} k/2} \ .$$

Using Hoeffding's third inequality instead of Azuma-Hoeffding's inequality [3], as in [22], leads to a better bound because the former is less general and tighter in our context.

The Q^* method enables us to recover the entire topology of T iff all r4-trees of Q_T are correctly inferred. As a consequence of lemma 4, this event occurs when $L_\infty(D^T,\hat{D}) < min_{e\in E_T} \frac{l(e)}{2}$. Thus, this requires that the $\binom{n}{2}$ distances in $\hat{D}$ be sufficiently close to those induced by T. Then, from lemma 5, we obtain:

Theorem 2 *Under the Cavender-Farris model of evolution, the probability that the Q^* method recovers the entire topology of an unknown tree T is at least*

$$1 - n^2 e^{-f^2 e^{-4d} k/2}$$

where $f = min_{e \in E_T} l(e)$ (assuming f close to 0). Equivalently, if we suppose k characters evolve on a phylogeny T under the Cavender-Farris model, then $T^* = T$ with probability at least $1-\epsilon$ $(\epsilon > 0)$ if

$$k > \frac{2 \ln(\frac{n^2}{\epsilon}) e^{4d}}{f^2} \quad .$$

It appears that the difficulty comes from short edges. However, since the inference of the different edges is independent for the Q^* method, we can obtain the following result in the same way as above:

Theorem 3 *Under the Cavender-Farris model of evolution, the probability that the Q^* method recovers an edge e is at least*

$$1 - n^2 e^{-l(e)^2 e^{-4d} k/2}.$$

The same property holds for the *Addtree* method [45] but not for the *Neighbor Joining* (*NJ*) method [44], since in this method, the inferred edges are interdependent. However, if we consider retrieving the whole structure of the unknown phylogeny, the same bound as obtained on the convergence rate of the Q^* method (theorem 2) can be shown for *NJ* and *Addtree*. This bound is better by a constant factor than the bound on the convergence rate of the *SP* method [1] (as used by Farach and Kannan [23]). It differs from the bound shown for the *SQM* method mainly because it depends on the diameter of the phylogeny, whereas the bound for *SQM* depends on the depth of the phylogeny. In the worst case, the diameter and the depth are of the same order, but for some distributions of trees (*e.g.*, the Yule-Harding distribution [31]), the depth is significantly less than the diameter with high probability. However, both measures are bounded by a small constant in practice. *E.g.*, Nei [42] recommends $\hat{D}_{xy} < 1$. A higher (but still small) constant can be considered if the evolutionary model takes rates heterogeneity into account. The reason for which biologists usually consider data sets with small $\hat{D}_{xy}$ values, is that high distances have a high variability, resulting in unprecise distance estimations, with the risk of leading any phylogenetic reconstruction method to infer erroneous trees. Thus, the difference between diameter and depth might be worthy in practice, but leads to bounds of the same order (see [4] for other reasons supporting this claim).

Considering d as a constant implies that the bound obtained on the convergence rate of all the above mentioned algorithms is $O(\frac{\log n}{f^2})$. f can be considered as varying in a sense opposite to n, because increasing n necessarily leads to decreasing f (newly added species break existing edges). In practice, biologists are confronted with problems resulting from the presence of small edges in the tree [25, 33, 48]. f can be very small, leading the above methods to require too high a number of characters to produce the correct tree with high probability. Moreover, it is improbable that a method will ever exist that can produce a reliable estimate of the complete phylogeny, in this area of the parameter space.

Finally, note that theorem 2 easily extends to more general stochastic models (cf. appendix). *E.g.*, for the generalized Jukes Cantor model [49] (enabling to consider molecular characters with 4, 20 or more states), we have $k > 2\frac{1}{f^2}\ln(\frac{n^2}{\epsilon})e^{2d/b}$. With four states, we have $b = 1 - (a^2 + b^2 + c^2 + d^2)$, where a (resp. c, g, t) is the probability of state A (resp. C,G,T) at the root. In that framework, the result for the original Jukes and Cantor model [34] is obtained with $b = \frac{3}{4}$, and for the Cavender Farris model with $b = \frac{1}{2}$.

6 Experimental results

The previous section showed theoretical worst case guarantees for the Q^* method. Here we focus on experimental results, to give insights as to the usefulness of the Q^* method for reconstructing phylogenies in practice.

6.1 Real data

The condition for edges to be in T^* is that all r4-trees they induce are 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 a high evolutive noise, it is likely that the Q^* method will only detect part of the inter-species relationships. However, most biological data sets do contain information that can be *extracted* by the Q^* method, as shown by the trees we obtained by applying the IQ^* algorithm to several real data sets taken from the literature: we obtained at least partially resolved trees (*i.e.*, no star tree) and even some fully resolved trees. As an example, we present in Fig. 6 the tree obtained for 11 mammals from a data set provided by D. Penny [43]. The data consists of DNA sequences of 191 nucleotides, obtained from several genes (α- and β-hemoglobins, fibrinopeptides A and B, cytochrome c, myoglobins, α-chrystallin). The r4-tree set Q was obtained by running *FPM* on distances collected from the sequences, corrected according to the Jukes Cantor model of evolution [34] (for this purpose we used the DNADIST program of the PHYLIP package [27]). 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, such as NJ (see [43]) and *Maximum Parsimony* (as can be checked by running existing software [27]), may be suspected of being incorrect, but Penny *et al* [43] report that they did not reach any firm conclusion concerning that edge. The Q^* relation takes no decision concerning the position of the rodent and the rabbit, in relation to the other mammals. To date, there 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 [21, 29].

Running times of the method are reasonable. For example, a non-optimized version of IQ^* required 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, required $0.07s$.

6.2 Simulated data

To further investigate the performances of the Q^* method, we performed simulations under various conditions of evolution, along the lines of [11, 37]. We generated rooted phylogenies by randomly chosing their structure from the Yule-Harding distribution [2, 31] and fixing their edge-lengths according to a Poisson model. The evolution of molecular sequences was then simulated along the edges of these phylogenies, from the root to the leaves, according to a given condition of evolution (fast evolutionary rates on all edges, medium rates on all edges, slow rates on all edges, fast/slow rates on half of the edges, fast/slow rates on half of the sites) and to the Kimura 2-parameter model of evolution [36]. This model was applied with a transition/transversion rate of 2.0. For each run, a data set was made up by taking the sequences obtained at the leaves of the phylogeny and converting them into a distance matrix, which was then corrected according to the Kimura model. In this way, we generated $25,000$ data sets on 10 species for the five conditions of evolution previously defined. We applied the Q^* method (*i.e.*, the *FPM* method, then the IQ^* algorithm) independently to each data set, measuring each time the number of incorrect edges inferred by the method (which may be seen as *false positives*), as well as the size of the tree T^* it output. The same process was applied to the NJ method. This method always inferred a fully resolved tree, so that each time it inferred a wrong edge, it forgot a correct edge (which may be seen as a false negative).

Depending on the condition of evolution, the sequence length and the data set, some edges of the model phylogeny T did not support any mutation. As a result, data sets did not always contain information for each edge of T (which mimics to some extent real situations). In these cases, the reconstruction method had no support to infer the corresponding edges. To account for this phenomenon, we also measured, for each data set, the number e_R of *realized* internal edges, *i.e.*, internal edges of the phylogeny which supported at least one substitution [38].

Table 1 displays, the experimental results obtained for the various conditions of evolution and sequence lengths.

Results confirm that the Q^* method usually produces trees which possess almost only safe edges. More precisely, it induced less than one wrong edge in ten trees ($\approx$ 1.3% incorrect edges) on average over all conditions of evolution. Even for the most difficult condition considered, *i.e.*, unequal rates of evolution among different sites (which violates an assumption of the Kimura model and thus lowers the accuracy of the distance corrections), the Q^* method only induced $\approx$ 3.9% incorrect edges on average. As a consequence of inferring almost only safe edges, Q^* usually produces trees which are to some extent partially resolved ($e_{T^*} < 100\%$). This implies that some correct edges were not inferred. However, less than 1/3 of the correct edges were missing on average (the percentage of false negatives are obtained from Table 1 by $1 - \%e_{T^*} + \%e_{fp}$). Moreover, we can see from the table that there is a real correlation between $\%e_R$ and $\%e_T$, meaning that the Q^* method does not try to randomly resolve edges for which the data set does not contain any information.

This behavior contrasts with that of most other methods, which infer fully resolved trees but usually with a non-negligible percentage of unsafe edges. *E.g.*, in the simulations, the NJ method always inferred fully resolved trees, containing on average more than one wrong edge in a tree, *i.e.*, $\approx$ 15.3% incorrect edges. The reason why usual methods infer fully resolved trees lies in the objective criterion they optimize: its value can always be improved by adding a new edge to the constructed tree. Thus, the resulting tree usually contains some edges specific to the data set rather than from the species' history. Biologists are aware of this overfitting problem from several studies showing a high variability observed within trees obtained by different methods on the same piece of data [26, 37], or when slightly varying the set of studied species [39, 43]. The Q^* method is one of the few methods which tries to avoid this overfitting effect (see [9] for other methods designed in that sense).

Because of their different purposes, it is difficult to compare the Q^* method to the usual reconstruction methods on the basis of their total error (*i.e.*, accounting for both false positives and false negatives). When giving the same cost to false positives and false negatives, we observed that NJ was on average better than Q^* in 7 conditions of evolution. However, in practice false positives are given much more importance, and giving them only twice as much importance as the false negatives (which can be thought as a minimum) leads the Q^* method to outperform the NJ in 9 conditions over 10.

7 Conclusion

We proposed a new quartet method, called Q^*, to reconstruct phylogenies. This method has the specificity to infer trees containing only combinatorially safe edges. As a result, this method is unlikely to produce incorrect edges (as confirmed by the experimental study). This suits the requirements of most biologists well, as they prefer having partially resolved trees with safe edges, rather than fully resolved trees with a non-negligible number of unsafe edges, as usually proposed by other methods. Unsafe edges greatly limit the confidence in the proposed tree, as all the inferred edges are usually interdependent. Note that this is not case in the Q^* method, where each edge only depends on the data r4-trees.

The objective criterion on which the Q^* method relies (maximum tree-like subset of r4-trees) can be exactly optimized in $O(n^4)$, where n is the number of species. This again contrasts with most of the usual criteria used to reconstruct phylogenies, which are NP-hard to optimize [19, 20, 46]. It is unlikely that we can improve the $O(n^4)$ time complexity for computing Q^* in the general case, *i.e.*, when the input is an arbitrary set of resolved quartets. However, if we consider as input a dissimilarity matrix $\hat{D}$ on the species, we might hope to lower the above complexity due to the fact that $\hat{D}$ contains only $O(n^2)$ information, from which we can infer all the resolved quartets by a distance principle.

Simulations showed that the Q^* method usually produces a partially resolved tree. If one aims at a more resolved tree, one can still consider T^* as a safe basis to which new edges should be added, *e.g.*, see [9, 12]. Warnow also recently showed the success of this approach [50] by completing T^* with the compatible edges of the NJ tree. The good results obtained through this practice [12, 50] also show that the Q^* method infers a non-negligible number of "non-trivial" edges, *i.e.*, edges which are not recovered by more traditional methods. This enlights another aspect of its usefulness. Moulton and Steel [41] also proposed completing the tree T^*, by considering a refinement of the Buneman relation (to which this T^* is equivalent) which can be computed in polynomial time [10, 13]. There is now some need of an experimental study to compare the various methods proposed for completing T^*.

Another topic worth exploring is the case where the input r4-tree set can contain several resolutions for some quartets (*e.g.* as with the ordinal r4-tree inference method [35]). In this paper, we chose to remove such r4-trees from the input data set, but designing an algorithm which can really handle such cases would be interesting.

Another issue to examine is whether the Q^* method may be of help to tackle in practice the NP-hard quartet consistency problem [46]. *E.g.*, the tree T^* might be a good basis for a branch-and-bound algorithm.

Finally, note that the source code and the executable of the Q^* method are available at the address `http://www.lirmm.fr/~berry`.

Acknowledgments

The authors thank H.-J. Bandelt for preliminary discussion, Tandy Warnow for kindly sending us a draft of the paper [22], Kevin Atteson for his help in deriving the proof of lemma 5, and D. Penny for providing the mammal data set. We are also grateful to the referees for their helpful comments.

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 ACM-SIAM SODA*, 1996.

[2] D.J. Aldous. *Mathematics and its application*, volume 76 of *IMA*, chapter Probabibility distributions on cladograms, pages 1–18. Springer Verlag, 1995.

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

[4] A. Ambainis, R. Desper, M. Farach, and S. Kannan. Nearly tight bounds on the learnability of evolution. In *IEEE Symposium on Foundations of Computer Science*, 1998.

[5] K. Atteson. The performance of neighbor-joining algorithms of phylogeny reconstruction. In *Proc. of COCOON*, Computing and Combinatorics, pages 101–110. Springer, 1997.

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

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

[8] J.P. Barthélemy and A. Guénoche. *Trees and proximities representations.* Wiley, 1991.

[9] V. Berry. *Méthodes et Algorithmes pour reconstruire les arbres de l'Évolution.* PhD thesis, Université de Montpellier - France, 1997.

[10] V. Berry. Improving the bound for computing the refined buneman tree. manuscript, 1998.

[11] 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.

[12] V. Berry and O. Gascuel. Reconstructing phylogenies from resolved 4-trees. Technical Report 97076, LIRMM, 1997.

[13] D. Bryant and V. Moulton. A polynomial time algorithm for constructing the refined buneman tree. *Appl. Math. Lett.*, in press, 1998.

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

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

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

[17] V. Chepoi and B. Fichet. L_∞ approximation via subdominants. *Journal of Mathematical Psychology*, 1998. (to appear).

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

[19] W. H. E. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. In *Bulletin of Mathematical Biology*, volume 49, pages 461–467. 1987.

[20] W.H.E. Day and D. Sankoff. Computational complexity of inferring phylogenies by compatibility. *Syst. Zool.*, 35(2):224–229, 1986.

[21] 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.

[22] P.L. Erdös, M.A. Steel, L.A. Szkely, and T.J. Warnow. Constructing big trees from short sequences. In *24th International Colloquium on Automata Langages and Programming*, 1997.

[23] M. Farach and S. Kannan. Efficient algorithms for inverting evolution. In *Proc. of the 28th Ann. ACM Symp. on Theory of Computing*, pages 230–236, 1996.

[24] James S. Farris. A probability model for inferring evolutionary trees. *Syst. Zool.*, 22:250–256, 1973.

[25] J. Felsenstein. Cases in which parsimony and compatibility methods will be positively misleading. *Syst. Zool.*, 27:401–410, 1978.

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

[27] J. Felsenstein. Phylip (phylogeny inference package) version 3.5c. Department of genetics, University of Washington, Seattle., 1993.

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

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

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

[31] E.F. Harding. The probabilities of rooted tree shapes generated by random bifurcation. *Adv. Appl. Math.*, 3:44–77, 1971.

[32] W. Hoeffding. Probability inequalities for sums of bounded random variables. *J. Am. Stat. Assoc.*, 58:13–30, 1963.

[33] D. M. Hillis J. P. Huelsenbeck. Success of phylogenetic methods in the four-taxon case. *Syst. Biol.*, 42(3):247–264, 1993.

[34] T.H. Jukes and C.R. Cantor. Evolution of protein molecules. In H. N. Munro, editor, *Mammalian Protein Metabolism*, volume III, chapter 24, pages 21–132. Academic Press, New York, 1969.

[35] P. Kearney. Ordinal beats additive. In *Second Annual International Conference on Computational Molecular Biology (RECOMB)*, 1998.

[36] M. Kimura. A simple method for estimating evolutionary rates base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*, 16:111–120, 1980.

[37] 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.

[38] S. Kumar. A stepwise algorithm for finding minimum evolution trees. *Mol. Biol. Evol.*, 13:584–593, 1996.

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

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

[41] V. Moulton and M. Steel. Retractions of finite distance functions onto tree metrics. *Disc. Appl. Math.*, 1998. (To appear).

[42] 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.

[43] 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.

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

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

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

[47] 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.

[48] D.L. Swofford, G.J. Olsen, P.J. Wadell, and D.M. Hillis. *Molecular systematics (2nd edition)*, chapter Phylogenetic Inference, pages 407–514. Sunderland, USA, 1996.

[49] F. Tajima and M. Nei. Estimation of evolution distance between nucleotide sequences. *Mol. Biol. Evol.*, 1(3):269–285, 1984.

[50] T. Warnow. personal communication, 1997.

8 Appendices

8.1 Proof of theorem 1

The IQ^* algorithm uses few data structures. The array Ext, addressed by the various component, indicates for each one if it can be extended from the current step i to the next one. For each component σ we also store the species it contains in a set X_σ, implemented as a chained list. The L_σ sets are coded as binary arrays of size n, indicating for each species if it belongs to σ and in the same time to an r4-tree required by σ but lacking in Q. Bipartitions of B_i^* are also stored as binary arrays.

Algorithm 2:

- The *initializations* require only constant time.
- At each of the $O(n)$ steps, the X_σ sets are initialized (line 1) by a simple recursive search of the tree T_{i-1}^*. This requires each time $O(n^2)$, proportional to the information to be stored, and thus $O(n^3)$ over the whole algorithm.
- One postordered search and a preordered one are initiated at each step of the main loop. The costs of these searches are detailed below (Algorithm 3).
- The set B_i^* is initialized in $O(n)$ by creating the bipartition $\{i\}|[i-1]$. From the Ext array we know in $O(1)$ if each of the $O(n)$ components of bipartitions of B_{i-1}^* can be extended (**line** 2). In the worst case all bipartitions of B_{i-1}^* can be extended, but one of them at most can support the addition of species i in its both components: the edge on which i is attached to the tree (if not on a node r, cf Fig. 3). Thus, bipartitions of B_i^* are simply obtained by adding species i in $O(1)$ time to the $O(n)$ bipartitions of B_{i-1}^* and by making a copy of one of them at most in $O(n)$, before adding i to it. Thus obtaining B_i^* requires $O(n)$ at each step, *i.e.*, $O(n^2)$ globally.
- The tree T_i^* is obtained in linear time from B_i^* by using Meacham's (1981) or Gusfield's (1991) algorithm.

Algorithm 3:

The tree T_{i-1}^* contains $O(n)$ edges, each one being processed twice at step i: once during the preordered search and once during the preordered one. Thus, from the global standpoint, $O(n^2)$ edges e_σ are considered over the whole algorihtm, *i.e.*, Algorithm 3 is performed $O(n^2)$ times. We now consider the various operations performed during this algorithm.

line 3: Emptying L_σ when a leaf is encountered in the tree (line 3) only costs $O(n)$, thus these operation is in time $O(n^3)$ over the whole algorithm.

line 4: $O(n^2)$ recursive calls are issued.

line 5: The initialization of L_σ is done in $O(n^2)$ since each union requires $O(n)$. Since algorithm 3 is performed $O(n^2)$ times, this operation globally costs $O(n^4)$.

lines 6-8: This set of operations consists in examining once each of the $3\binom{n}{4}$ possible r4-trees (as established by Lemma 3). Issuing the request to Q for each r4-tree is done in time $O(1)$ from the X_σ sets (coded as chained lists). Moreover, knowing if a given r4-tree is contained in Q (line 8) is immediate since we reasonnably assume that Q is stored as a four-dimentionned array (if not, an $O(n^4)$ simple preprocessing insures it). This array indicates for each quartet of species the corresponding r4-tree present in Q (or the absence of r4-tree). Adding a species x to L_σ is also immediate.

As a result, lines 6-8 globally require time $O(n^4)$. Moreover, the leading coefficient is small since $3\binom{n}{4} < c.n^4$, with $c < 1$.

line `9`: Deciding whether a component can be extended requires the complete examination of L_σ, *i.e.*, $O(n)$. Ext_σ is set in $O(1)$. Thus, over the $O(n^2)$ executions of Algorithm 3, this line requires time $O(n^3)$.

The above analysis shows that IQ^* computes T^* and B^* in time $O(n^4)$, steps `5` and `6-8` being the most time-consuming.

□

8.2 Proof of lemma 5

For the sake of clarity, we denote D^T_{xy} by D, $\hat{D}_{xy}$ by $\hat{D}$, p_{xy} by p and f_{xy} by f. We give the proof for the generalized Jukes Cantor model [49], where we have

$$D = -b\ln(1-p/b) \;\text{ and }\; \hat{D} = -b(1-f/b) \ ,$$

where

$$b = 1 - \sum_{s\in X} p(s)^2 \ ,$$

X being the set of all states that considered characters can take (*e.g.*, $X = \{A, C, G, T\}$ for nucleotide sequences) and $p(s)$ being the probability of observing state s (the process is assumed to be at equilibrium). An important property used below is that the function $-b\ln(1-p/b)$ is strictly increasing.

To prove the lemma, we first concentrate on the event

$$|D - \hat{D}| \quad \geq \quad \epsilon \ . \tag{1}$$

We separately analyze the two cases:

1. <u>$D > \hat{D}$ (and $p > f$)</u>:

$$\begin{aligned} |D-\hat{D}| \geq \epsilon \quad &\equiv \quad D - \hat{D} \geq \epsilon \\ &\equiv \quad b\ln\left(\frac{b-p}{b-f}\right) \leq -\,\epsilon \\ &\equiv \quad 1 - \frac{p-f}{b-f} \leq e^{-\epsilon/b} \\ &\equiv \quad p - f \geq (1-e^{-\epsilon/b})(b-f) \\ &\Rightarrow \quad p - f \geq (1-e^{-\epsilon/b})(b-p) \ . \end{aligned} \tag{2}$$

2. <u>$D < \hat{D}$ (and $f > p$)</u>:

$$\begin{aligned} |D-\hat{D}| \geq \epsilon \quad &\equiv \quad \hat{D} - D \geq \epsilon \\ &\equiv \quad b\ln\left(\frac{b-f}{b-p}\right) \leq -\,\epsilon \\ &\equiv \quad 1 + \frac{p-f}{b-p} \leq e^{-\epsilon/b} \\ &\equiv \quad f - p \geq (1-e^{-\epsilon/b})(b-p) \ . \end{aligned} \tag{3}$$

From (2) and (3) we have:

$$\begin{aligned} |D-\hat{D}| \geq \epsilon \quad &\Rightarrow \quad |p-f| \geq (1-e^{-\epsilon/b})(b-p) \\ &\Rightarrow \quad |p-f| \geq b(1-e^{-\epsilon/b})e^{-D/b} \geq b(1-e^{-\epsilon/b})e^{-d/b} \ , \end{aligned} \tag{4}$$

(since $d = \underset{xy}{max} D^T_{xy}$).

Hoeffding's third inequality [32] states that:

$$P\left(|p-f| \geq g\right) \;\leq\; 2\, e^{-2kg^2} \;.$$

We use this inequality in the following way:

$$P\left(|p-f| \geq b(1-e^{-\epsilon/b})e^{-d/b}\right) \;\leq\; 2\, e^{-2kb^2(1-e^{\epsilon/b})^2 e^{-2d/b}} \;. \tag{5}$$

From (4) and (5) we then have

$$P(|D-\hat{D}| \geq \epsilon) \;\leq\; 2\, e^{-2kb^2(1-e^{-\epsilon/b})^2 e^{-2d/b}} \;,$$

which gives the result:

$$P(|D-\hat{D}| < \epsilon) \;\geq\; 1 - 2\, e^{-2kb^2(1-e^{-\epsilon/b})^2 e^{-2d/b}} \;.$$

For example, for the Cavender Farris model, where we have $b = \frac{1}{2}$, this gives

$$P(|D-\hat{D}| < \epsilon) \;\geq\; 1 - 2\, e^{-\frac{1}{2}k(1-e^{-2\epsilon})^2 e^{-4d}} \;.$$

□

Algorithm 1: Deciding whether a component σ of a $B^*_{[i-1]}$ 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 $x \in \sigma$, $y \in \overline{\sigma}_j$, $z \in \overline{\sigma}_k$ ($\equiv xi|yz \in R^{j,k}_\sigma$)
        if $xi|yz \notin Q$ then $L_\sigma \leftarrow L_\sigma \cup \{x\}$
if $L_\sigma = \emptyset$ then answer YES else answer NO.
```

Algorithm 2: Main part of IQ^*.

```
Input  : A set $Q$ of r4-trees defined on species $1, 2, \ldots, n$.
Output: The tree $T^* \equiv Q^*$.

/* Initializations */

$B^*_{[4]} \leftarrow \{ \{1\}|\{2,3,4\}, \{2\}|\{1,3,4\}, \{3\}|\{1,2,4\}, \{4\}|\{1,2,3\} \}$
if $\exists\, r = xy|zt$, $\{x,y,z,t\} = \{1,2,3,4\}$ and s.t. $r \in Q$ then
    $B^*_{[4]} \leftarrow B^*_{[4]} \cup \{ \{x,y\}|\{z,t\} \}$ , $T^*_{[4]} \leftarrow$ topology of $r$
else $T^*_{[4]} \leftarrow$ star topology on species $\{1,2,3,4\}$

/* Progressively insert the remaining species */

foreach step $i \leftarrow 5$ to $n$
    foreach subtree $\sigma \in T^*_{[i-1]}$ do compute the set $X_\sigma$ of species contained in $\sigma$
    Let $e$ be the edge incident to the root of $T^*_{[i-1]}$
    PostOrder($e$) /* process components associated with upper subtrees of $T^*_{[i-1]}$ */
    PreOrder($e$)  /* process components associated with lower subtrees of $T^*_{[i-1]}$ */

    /* Deduce $B^*_{[i]}$ and $T^*_{[i]}$ */

    $B^*_{[i]} \leftarrow \{ \{i\}|\{1, \ldots, i-1\} \}$
    foreach bipartition $\sigma|\overline{\sigma} \in B^*_{[i-1]}$
        if $Ext_\sigma$ then $B^*_{[i]} \leftarrow B^*_{[i]} \cup \{\sigma \cup \{i\}|\overline{\sigma}\}$
        if $Ext_{\overline{\sigma}}$ then $B^*_{[i]} \leftarrow B^*_{[i]} \cup \{\sigma|\overline{\sigma} \cup \{i\}\}$
    Construct $T^*_{[i]}$ from $B^*_{[i]}$
```

Algorithm 3: $PostOrder(e_\sigma)$: performs a postordered search of the lower subtree $\overline{\sigma}$ incident to edge e_σ. This enables us to know if the component σ associated with the upper subtree of e_σ can contain species i, *i.e.*, if $\sigma|\overline{\sigma} \in B^*_{[i-1]}$ can be extended into $\sigma \cup \{i\}|\overline{\sigma} \in B^*_{[i]}$.

```
if the lower subtree induced by $e_\sigma$ is a leaf then $L_\sigma \leftarrow \emptyset$
else
    /* Process the components which contain $\sigma$ */
    foreach $e_j, j \in 1..l$, child edge of $e_\sigma$ in $T^*_{[i-1]}$ do PostOrder($e_j$)
    /* Process $\sigma$ */
    /* 1: reuse results obtained for r4-trees $r \in R_\sigma - R^c_\sigma$ */
    $L_\sigma \leftarrow \bigcup_{j \in 1..l} L_j \cap X_\sigma$
    /* 2: examine r4-trees $r \in R^c_\sigma$ */
    foreach pair $j, k \in 1..l$ of child edges of $e_\sigma$ do
        /* examine r4-trees of $R^{j,k}_\sigma$ */
        foreach r4-tree $xi|yz$ s.t. $x \in X_\sigma$, $y \in X_{\sigma_j}$, $z \in X_{\sigma_k}$ do
            if $r \notin Q$ then
                $L_\sigma \leftarrow L_\sigma \cup \{x\}$    /* keep track of species belonging to $\sigma$ and to an
                                                  r4-tree responsible for the non-extension of $\sigma$ */
    /* Decide whether component $\sigma$ can be extended */
    if $L_\sigma = \emptyset$ then
        $Ext_\sigma$ = YES    /* all r4-trees required by $\sigma$ to include $i$ are present in $Q$ */
    else $Ext_\sigma$ = NO    /* some r4-trees required by $\sigma$ are lacking in $Q$ */
```

Table 1: Experimental results for the Q^* and NJ method

rates	#sites	$\% e_{fp}(NJ)$	$\% e_{fp}(Q^*)$	$\% e_{T^*}$	$\% e_R$
slow	300	35.14	1.00	54.7	58.6
	1000	14.00	0.71	76.9	82.4
medium	300	11.71	0.57	72.4	88.7
	1000	5.29	0.14	84.7	96.5
fast	300	8.86	0.28	75.1	95.0
	1000	3.57	0.14	86.4	98.7
fast/slow per edge	300	22.14	1.42	59.7	78.3
	1000	12.43	1.14	71.0	90.8
fast/slow per site.	300	22.29	3.43	59.4	95.1
	1000	18.29	4.43	70.9	98.4

Note to Table 1: results (averaged over $25,000$ data sets on 10 species) obtained by the Q^* and the NJ method for reconstructing a phylogeny under various conditions of evolution for sequences of 300 and 1000 sites. The following conditions of evolution were investigated: *slow* rates (≈ 0.02 expected number of mutations per site from root to a leaf), *medium* rates (≈ 0.1 expected mutations), *fast* rates (≈ 0.2 expected mutations), *fast/slow per edges* (fast rate on half of the edges and slow rate on the others), and *fast/slow per site* (fast rate on half of the sites, slow rate on the others). $\% e_{fp}(NJ)$ and $\% e_{fp}(Q^*)$ express (in percent) the number of incorrect edges inferred by the corresponding method divided by the number of internal edges of the correct phylogeny T (*i.e.*, 7). $\% e_{T^*}$ is (in percent) the number of internal edges contained in the tree inferred by the Q^* method divided by the number of internal edges of T. $\% e_R$ is (in percent) the number of internal edges of the model phylogeny where mutations actually occured when generating a data set, divided by the number of internal edges of T.

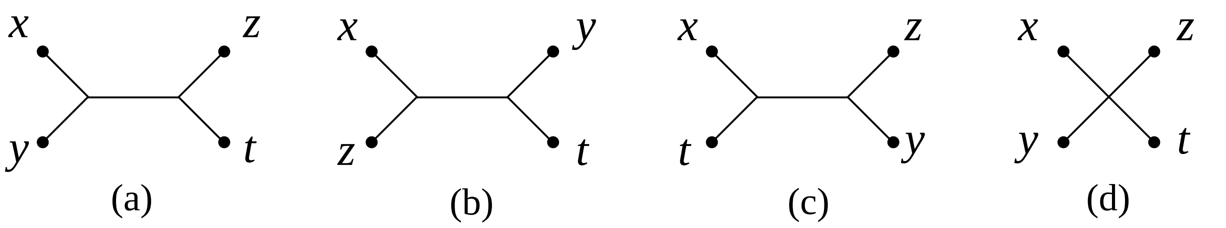

Figure 1: The four possible subtrees on four species: (a-c) the three r4-trees; (d) the star topology.

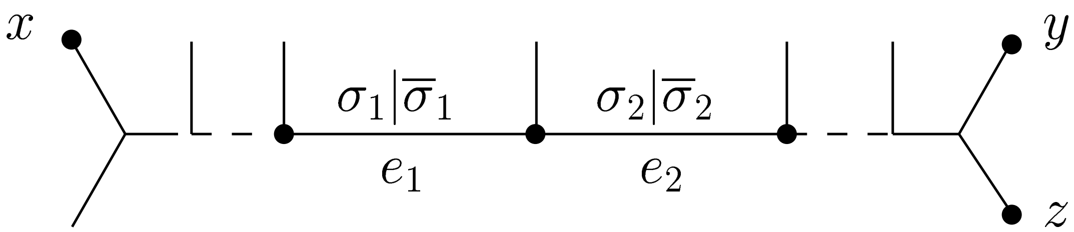
x
$\sigma_1|\overline{\sigma}_1$
$\sigma_2|\overline{\sigma}_2$
y
e_1
e_2
z

Figure 2: Containment relations between components.

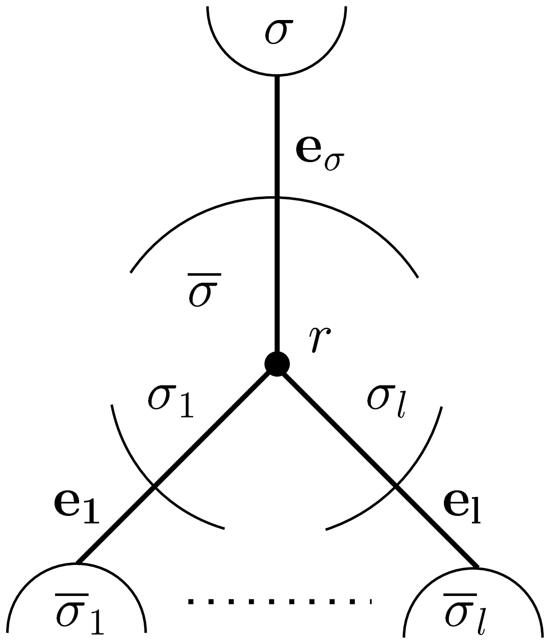

Figure 3: Situation arising around an internal node.

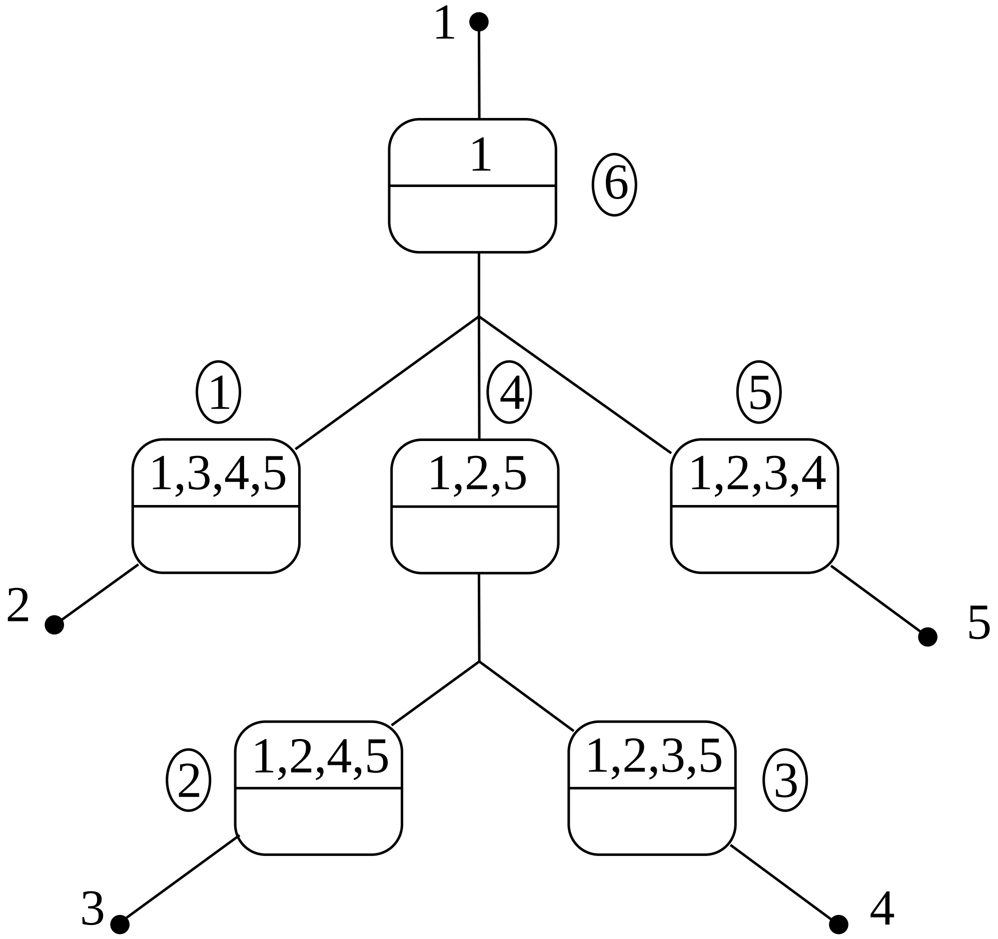

Figure 4: Postordered depth-first search processing the upper subtrees (circled numbers indicate their processing order). Information concerning the upper subtrees is passed from the leaves to the root. Subtrees are labelled with the information resulting from their processing.

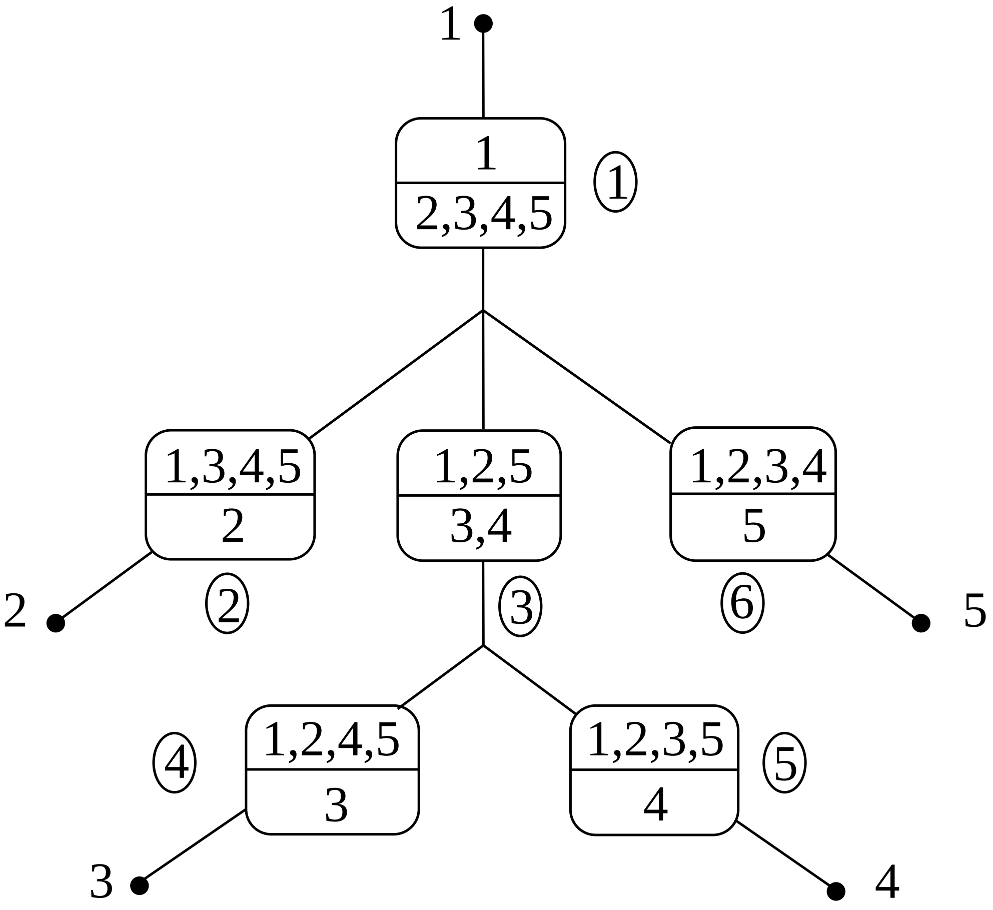

Figure 5: Preordered depth-first search processing the lower subtrees (circled numbers indicate their processing order). Information concerning the lower subtrees is passed from the root to the leaves.

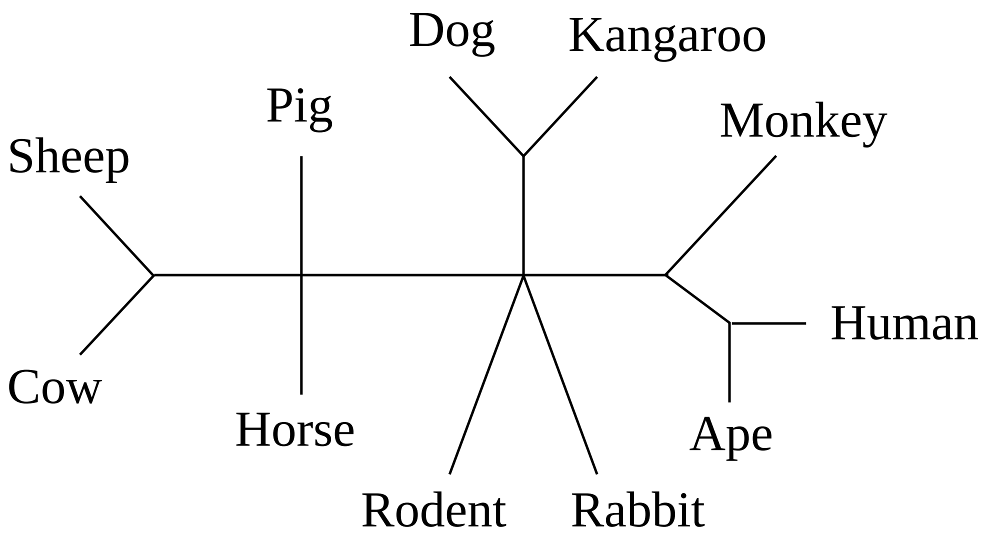

Figure 6: Applying Q^* to 11 mammal sequences