

A quadratic time algorithm for computing the quartet distance between two general trees

Thomas Mailund, Jesper Nielsen and Christian N.S. Pedersen

Bioinformatics Research Centre
Aarhus University, Aarhus, Denmark
Email: {mailund,jn,cstorm}@birc.au.dk

Abstract

We derive a quadratic time and space algorithm for computing the quartet distance between a pair of general trees, i.e. trees where inner nodes can have any degree ≥ 3 . The time and space complexity of our algorithm is quadratic in the number of leaves and does not depend on the degree of the inner nodes. This makes it the fastest algorithm for computing the quartet distance between general trees independent of the degree of the inner nodes.

1. Introduction

The evolutionary relationship between a set of species is conveniently described as a tree, where the leaves represent the species and the inner nodes speciation events. Using different inference methods to infer such trees from biological data, or using different biological data from the same set of species, often yield slightly different trees. To study such differences in a systematic manner, one must be able to quantify differences between evolutionary trees using well-defined and efficient methods. One approach for this is to define a distance measure between trees and compare two trees by computing this distance. Several distance measures have been proposed, e.g. the symmetric difference metric [7], the nearest-neighbour interchange metric [11], the subtree transfer distance [1], the Robinson and Foulds distance [8], and the quartet distance [6]. Each distance measure has different properties and reflects different aspects of biology.

In this paper, we derive an $O(n^2)$ time and space algorithm for computing the quartet distance between a pair of trees. For an evolutionary tree, the *quartet topology* of four species is determined by the minimal topological subtree containing the four species. The four possible quartet topologies of four species are shown in Fig. 1. Given two evolutionary trees on the same set of n species, the *quartet distance* between them is the number of sets of four species for which the quartet topologies differ in the two trees.

Most of previous work has focused on comparing *binary* trees and therefore avoided star quartets. Steel and Penny in [9] developed an algorithm for computing the quartet distance in time $O(n^3)$. Bryant *et al.* in [3] improved this

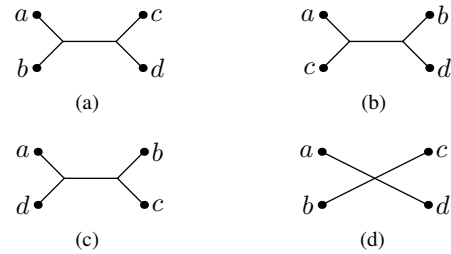


Figure 1. The four possible quartet topologies of species a , b , c , and d . Topologies (a): $ab|cd$, (b): $ac|bd$, and (c): $ad|bc$ are *butterfly* quartets, while topology (d): $\frac{a}{b} \times \frac{c}{d}$, is a *star* quartet. For binary trees, only the butterfly quartets are possible.

result with an algorithm that computes the quartet distance in time $O(n^2)$. Brodal *et al.*, in [2], presented the currently best known algorithm that computes the quartet distance in time $O(n \log n)$.

Recently, we have developed algorithms for computing the quartet distance between two trees of *arbitrary* degrees, i.e. trees that can contain star quartets. In [4] we developed two algorithms: the first algorithm runs in time $O(n^3)$ and space $O(n^2)$ —and is thus independent of the degree of the inner nodes—the second in time $O(n^2 d^2)$ and space $O(n^2)$, where d is the maximal degree of inner nodes in the trees—and thus depend on the degree of the nodes. The $O(n^2 d^2)$ was later improved to $O(n^2 d)$ [5] and by taking an approach similar to the Brodal *et al.* [2] $O(n \log n)$ we developed a sub-quadratic algorithm in terms of n but at a significant cost in terms of d : $O(d^9 n \log n)$ [10].

In this paper we develop an $O(n^2)$ time and space algorithm, where the running time is independent of the degrees of the inner nodes of the input trees.

2. Background

The quartet distance between two trees is the number of quartets where the quartet topology differ between the two trees, i.e. the number of quartets where one tree has the star topology and the other a butterfly topology, plus the number of quartets where both trees have a butterfly topology, but

different ones. As observed in [4], the former—where one tree has the star topology and the other a butterfly—can be expressed in terms of the total number of butterflies in the two trees, the number of shared butterflies and the number of different butterflies: For trees T and T' , the number of different topologies due to one being a star and the other a quartet, $\text{diff}_S(T, T')$, is given by

$$\text{diff}_S(T, T') = B + B' - 2(\text{shared}_B(T, T') + \text{diff}_B(T, T')) , \quad (1)$$

where B is the number of butterflies in T , B' the number of butterflies in T' , $\text{shared}_B(T, T')$ the number of quartets with the same butterfly topology in T and T' and $\text{diff}_B(T, T')$ the number of quartets with different butterfly topologies in T and T' . Thus the quartet distance between T and T' is given by the expression

$$\text{qdist}(T, T') = B + B' - 2\text{shared}_B(T, T') - \text{diff}_B(T, T') . \quad (2)$$

Since, $B = \text{shared}_B(T, T)$ and $B' = \text{shared}_B(T', T')$, an algorithm for computing $\text{shared}_B(T, T')$ and $\text{diff}_B(T, T')$ gives an algorithm for computing the quartet distance between T and T' .

Our approach to counting the shared and different quartets is based on *directed quartets* and *claims* [2], [4]. An (undirected) butterfly quartet topology, $ab|cd$ induces two directed quartet topologies $ab \rightarrow cd$ and $ab \leftarrow cd$, by the orientation of the middle edge of the topology, as shown in Fig. 2. There are twice as many directed butterflies as undirected, and the number of shared (different) butterflies can be counted as half the number of shared (different)

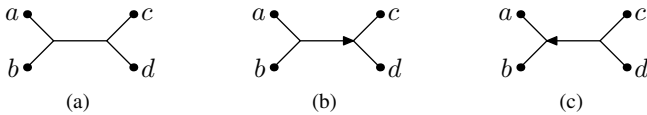


Figure 2. An undirected quartet topology, (a), and the two directed quartet topologies, (b) and (c), induced by it.

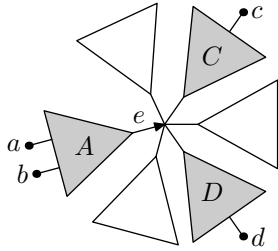


Figure 3. A *claim* $A \xrightarrow{e} (C, D)$. The claim $A \xrightarrow{e} (C, D)$ claims all ordered butterflies $ab \rightarrow cd$ where $a, b \in A$ and $c \in C, d \in D$ where C and D are two *different* subtrees in front of e .

directed butterflies. To each directed quartet, $ab \rightarrow cd$, we can uniquely associate a directed edge, e such that a and b are leaves in the tree behind e , and such that c and d are leaves in *different* subtrees of the root of the tree in front of e , see Fig. 3. We call such a tree substructure, consisting of a directed edge e with a subtree, A behind e and two distinct subtrees, C and D , in front of e a *claim*, written $A \xrightarrow{e} (C, D)$, and say that the edge e *claims* the directed quartet $ab \rightarrow cd$ and we also say that an edge e claims an undirected quartet $ab|cd$ if it claims one of its directed quartets. Each (undirected) butterfly quartet defines exactly two directed butterfly quartets, and each directed quartet is claimed by exactly one directed edge; considering each claim and implicitly each directed butterfly claimed by the claim, we can examine each directed butterfly in a tree, or each undirected butterfly twice.

The crux of the algorithm is to consider each pair of claims, one from each tree, and for each such pair count the number of shared and different directed butterflies claimed in the two trees. Dividing these counts by two gives us $\text{shared}_B(T, T')$ and $\text{diff}_B(T, T')$.

3. A Quadratic Time and Space Algorithm

For a tree with n leaves, there are $m < n$ inner nodes and $n + m - 1$ edges. Assume we are given two nodes $v \in T$, $v' \in T'$ of degrees d_v and $d_{v'}$ respectively. Spending time $O(d_v d_{v'})$ on each pair of inner nodes v, v' , one from each tree, thus result in a total running time of

$$O\left(\sum_{v \in T} \sum_{v' \in T'} d_v d_{v'}\right) = O\left(\left(\sum_{v \in T} d_v\right) \left(\sum_{v' \in T'} d_{v'}\right)\right) \quad (3)$$

$$= O(n^2)$$

since $\sum_{v \in T} d_v$ and $\sum_{v' \in T'} d_{v'}$ both are $O(n)$.

3.1. Preprocessing

Before counting shared and different butterflies, we calculate a number of matrices in two preprocessing steps. First, we calculate a matrix that for each pair of subtrees $F \in T$ and $G \in T'$ stores the number of leaves in both trees, $|F \cap G|$. This can be achieved in time and space $O(n^2)$ [3].

Next, for each pair of inner nodes, $v \in T, v' \in T'$ with sub-trees $F_i, i = 1, \dots, d_v$ and $G_j, j = 1, \dots, d_{v'}$, respectively, we calculate a matrix, I , such that $I[i, j] = |F_i \cap G_j|$. We also calculate vectors of row and column sums and the total matrix sum:

$$R[i] = \sum_{j=1}^{d_{v'}} I[i, j] , \quad (4)$$

$$C[j] = \sum_{i=1}^{d_v} I[i, j] , \quad (5)$$

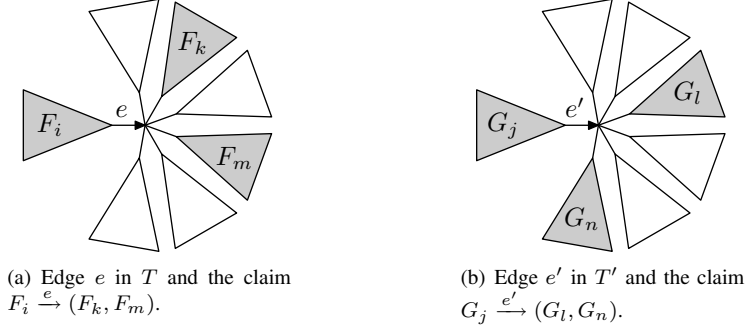


Figure 4. A pair of inner edges, $e \in T$, $e' \in T'$, where F_i (G_j) is the sub-tree behind e (e') and $F_k, k \neq i$ ($G_l, l \neq j$) the remaining subtrees of the node pointed to by e (e'). Highlighted are two claims, one from each tree.

$$M = \sum_{i=1}^{d_v} \sum_{j=1}^{d_{v'}} I[i, j]. \quad (6)$$

Calculating these values is done in time $O(d_v d_{v'})$ for each pair of inner nodes, giving a total preprocessing time of $O(n^2)$.

3.2. Counting shared butterfly topologies

For each pair of inner edges, $e \in T$ and $e' \in T'$, see Fig. 4, we count the directed butterflies claimed by both e and e' . These are all on the form $ab \rightarrow cd$ where $a, b \in F_i \cap G_j$, $c \in F_k \cap G_l$ and $d \in F_m \cap G_n$ for some claims, $F_i \xrightarrow{e} (F_k, F_m)$ and $G_j \xrightarrow{e'} (G_l, G_n)$, of e and e' . The total number of directed butterflies common for both e and e' is therefore given by the expression

$$\frac{1}{4} \binom{|F_i \cap G_j|}{2} \sum_{k \neq i} \sum_{l \neq j} |F_k \cap G_l| \sum_{m \neq i, k} \sum_{n \neq j, l} |F_m \cap G_n| \quad (7)$$

or the sum of $\frac{1}{4} \binom{|I[i, j]|}{2} \cdot I[k, l] \cdot I[m, n]$ for all distinct entries in I but fixed (i, j) , see Fig. 5(a). We divide by four since we count each quartet four times, due to symmetry between m and k and between n and l .

Notice, however, that the inner sum is simply the total sum of entries, M , except for the rows i and k and columns j and l , see Fig. 5(b). Using

$$\sum_{m \neq i, k} \sum_{n \neq j, l} |F_m \cap G_n| = \quad (8)$$

$$M - \sum_{q=i, k} R[q] - \sum_{r=j, l} C[r] + \sum_{q=i, k} \sum_{r=j, l} I[q, r]$$

we can thus, given the preprocessing, compute the inner sum in time $O(1)$. The entire expression in eq. (7) can then be computed in time $O(d_v d_{v'})$, and thus we can compute all shared directed butterflies in total time $O(n^2)$. Dividing by two, we get the number of shared undirected butterflies.

3.3. Counting different butterfly topologies

Counting the number of different butterflies in the two trees is done similar to counting the number of shared butterflies. As before, we consider a pair of inner edges, $e \in T$ and $e' \in T'$. The quartets claimed by both e and e' , but with different butterfly topology, are on the form $a \in F_i \cap G_j$, $b \in F_i \cap G_l$, $c \in F_k \cap G_j$ and $d \in F_m \cap G_n$ for some claims $F_i \xrightarrow{e} (F_k, F_m)$ and $G_j \xrightarrow{e'} (G_l, G_n)$. The number of butterflies claimed by both e and e' but with different topology is therefore given by

$$|F_i \cap G_j| \sum_{k \neq i} \sum_{l \neq j} |F_i \cap G_l| |F_k \cap G_j| \sum_{m \neq i, k} \sum_{n \neq j, l} |F_m \cap G_n| \quad (9)$$

or the sum of $I[i, j] \cdot I[i, l] \cdot I[k, j] \cdot I[m, n]$ for all distinct entries in I but fixed (i, j) , see Fig. 6(a). Note that in this case we do not need to divide by any normalizing constant, since there are no symmetries between k and m or between l and n .

As before, the inner sum can be expressed as in eq. (8) and thus eq. (9) can be computed in time $O(d_v d_{v'})$ giving a total time of $O(n^2)$ to compute different directed, and thus different undirected, butterfly topologies in the two trees.

4. Conclusions

We have presented an algorithm that computes the quartet distance between two general trees in time $O(n^2)$, i.e. where the running time is independent of the degree of inner nodes, unlike previous algorithms for general trees.

The algorithm computes a set of matrices and values associated with these—total sum, row and column sums—in a preprocessing step, which enable it to compute the number of shared and different directed butterfly quartets claimed by a pair of inner edges. The algorithm uses time $O(d_v d_{v'})$ for each pair of nodes $v \in T$ and $v' \in T'$, where d_v and $d_{v'}$ is the degree of nodes v and v' . Since the sums over d_v and $d_{v'}$ are $O(n)$, this leads to a total running time of $O(n^2)$.

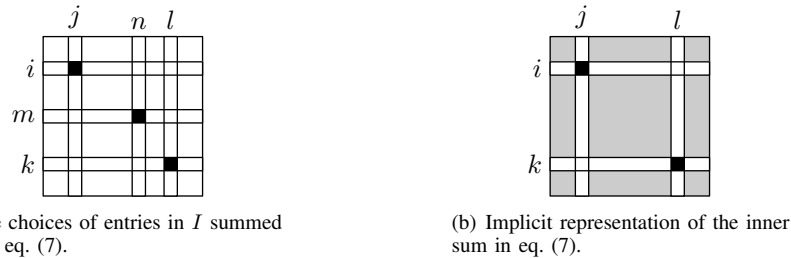


Figure 5. Graphical illustration of the shared quartet expression, eq. (7). On the left, the matrix entries summed over are explicitly shown. On the right, the inner sum is implicitly shown. The sum of the greyed entries can be computed in constant time.

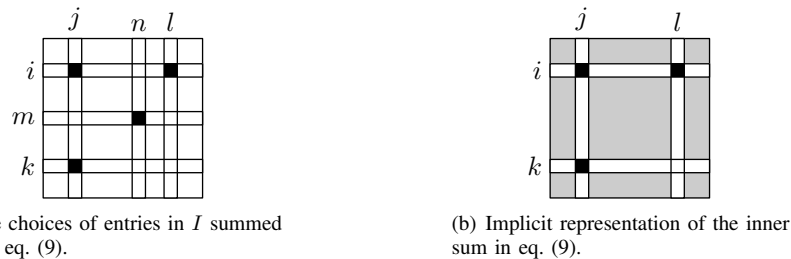


Figure 6. Graphical illustration of the different quartet expression, eq. (9). On the left, the matrix entries summed over are explicitly shown. On the right, the inner sum is implicitly shown. The sum of the greyed entries can be computed in constant time.

Acknowledgements

We are grateful to Chris Christiansen, Martin Randers and Martin S. Stissing for many fruitful discussions about the quartet distance.

References

- [1] B. L. Allen and M. Steel. Subtree transfer operations and their induced metrics on evolutionary trees. *Annals of Combinatorics*, 5:1–13, 2001.
- [2] G. S. Brodal, R. Fagerberg, and C. N. S. Pedersen. Computing the quartet distance between evolutionary trees in time $O(n \log n)$. *Algorithmica*, 38:377–395, 2003.
- [3] D. Bryant, J. Tsang, P. E. Kearney, and M. Li. Computing the quartet distance between evolutionary trees. In *Proceedings of the 11th Annual Symposium on Discrete Algorithms (SODA)*, pages 285–286, 2000.
- [4] C. Christiansen, T. Mailund, C. N. S. Pedersen, and M. Randers. Algorithms for computing the quartet distance between trees of arbitrary degree. In *Proc. of Workshop on Algorithms in Bioinformatics (WABI)*, volume 3692 of *Lecture Notes in Bioinformatics (LNBI)*, pages 77–88. Springer-Verlag, 2005.
- [5] C. Christiansen, T. Mailund, C. N. S. Pedersen, M. Randers, and M. S. Stissing. Fast calculation of the quartet distance between trees of arbitrary degrees. *Algorithms for Molecular Biology*, 1, 2006.
- [6] G. Estabrook, F. McMorris, and C. Meacham. Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units. *Syst. Zool.*, 34:193–200, 1985.
- [7] D. F. Robinson and L. R. Foulds. Comparison of weighted labelled trees. In *Combinatorial mathematics, VI (Proc. 6th Austral. Conf)*, Lecture Notes in Mathematics, pages 119–126. Springer, 1979.
- [8] D. F. Robinson and L. R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.
- [9] M. Steel and D. Penny. Distribution of tree comparison metrics—some new results. *Syst. Biol.*, 42(2):126–141, 1993.
- [10] M. Stissing, C. N. S. Pedersen, T. Mailund, G. S. Brodal, and R. Fagerberg. Computing the quartet distance between evolutionary trees of bounded degree. In Sankoff, D and Wang, L and Chin, F, editor, *Proceedings of the 5th Asia-Pacific Bioinformatics Conference 2007*, volume 5 of *Series on Advances in Bioinformatics and Computational Biology*, pages 101–110, 2007.
- [11] M. S. Waterman and T. F. Smith. On the similarity of dendrograms. *Journal of Theoretical Biology*, 73:789–800, 1978.