The XIII International Conference
"Applied Stochastic Models and Data Analysis" (ASMDA-2009)
June 30-July 3, 2009, Vilnius, LITHUANIA

ISBN 978-9955-28-463-5
L. Sakalauskas, C. Skiadas and E. K. Zavadskas (Eds.): ASMDA-2009
Selected papers. Vilnius, 2009, pp. 6–10

ABOUT THE LARGEST SUBTREE COMMON TO SEVERAL PHYLOGENETIC TREES

Alain Guénoche[1], Henri Garreta[2] and Laurent Tichit[3]

[1, 3]*IML-CNRS, 163 Av. de Luminy, 13009 Marseille*
[2, 3]*LIF-Université de la Méditerranée, 163 Av. de Luminy, 13009 Marseille*
E-mail: [1]*guenoche@iml.univ-mrs.fr*

Abstract: Given several phylogenetic trees on the same set of taxa X, we look for a largest subset Y in X such that all the partial trees reduced by Y are identical. The problem has polynomial complexity when there are only two trees but it is NP-hard for more than two. We introduce a polynomial approximation algorithm for the multiple case, which is easy to implement, very efficient and which produces a maximal common subtree. It begins with the computation of an upper bound for its size and designates elements in X that cannot belong to a common subtree of a given size. Simulations on random and real data have shown that this heuristic always provides an optimal solution as soon as the number of taxa is lower than 100. Then, we develop a statistical study to determine the critical size of a MAST to be significant, that is corresponding to non-independant trees.

Keywords: Phylogenetic tree, Common partial tree, MAST.

1. Introduction

This problem appears when comparing several X-trees connecting the same set of taxa X. Let us recall that an X-tree is a partially labeled tree such that (i) X is the set of labeled leaves, (ii) all the unlabeled nodes have degree at least 3 and (iii) the edges have positive or null length. They are unrooted, and when a root is placed on one edge, they become *phylogenetic trees*. These X-trees are computed from aligned sequences with a bootstrap strategy or when comparing the trees obtained from several genes. For these latter, the orthologous gene sequences in each taxon being determined, their comparison, with any reconstruction method (maximum likelihood, parsimony, distance method, etc.), gives a X-tree or a phylogenetic tree. Generally, different genes lead to different trees because of biological reasons such as the nucleotide composition, the evolution speed along the branches or the horizontal gene transfers. When considering p genes, one gets a set $\{T_1, T_2, \ldots, T_p\}$ of X-trees. The question is to study the compatibility of the T_i trees. Aside the Robinson-Foulds metric [Robinson and Foulds 1981], the compatibility can be measured by the size of a largest subset Y in X for which the trees agree, indicating the same evolution story.

The mathematical and computational study of X-trees has been established all along the last forty years. Over numerous articles, one can refer to the books of [Barthélémy and Guénoche 1991] and [Semple and Steel 2003]. In this latter the question is tackled in chapter 6 as the Maximum Agreement SubTree (MAST) problem. When there are only two rooted trees, [Steel and Warnow 1993] defined a dynamic programming scheme (hence a polynomial algorithm) which can be extended to the unrooted case. But when $p > 2$ the problem becomes NP-hard. More recently, [Cole *et al.* 1996], [Amir and Kesselman 1997] have proposed $O(n \log n)$ algorithms for two rooted binary trees and [Berry and Nicolas, 2004] give a $O(3^k p\, n \log n)$ algorithm for p binary trees, k being the number of elements to eliminate, to make a MAST.

In this article, we describe a method to establish a common subtree as large as possible, which can be applied to more than two binary or not binary unrooted trees, which is easy to program and very efficient on real problems up to $n = 100$, $p = 100$.

2. Methodology

Here, we are only interested in the *X*-tree shape, which is pompously called its *topology*, whatever the length of the edges are. In that case, these lengths can all be set to 1, and the path length distance in the tree receives integer values. It is a tree distance *D*, satisfying to the Four Point Condition:

$$\forall\{x,y,z,t\}, D(x,y)+D(z,t) \leq \max\{D(x,z)+D(y,t), D(x,t)+D(y,z)\}.$$

The distance values indicate any quadruple topology ; if, for $\{x,y,z,t\}$ $D(x,y)+D(z,t)$ is the smallest of the three sums, this quadruple has topology $xy|zt$ and, in the support tree of *D*, at least one edge separates these two pairs. If the three sums are equal, the topology is said to be *non resolved* and there is no separating edge.

To decide if two *X*-trees *A* et *B* have the same topology, it is sufficient to compare

(i) their distances D_A and D_B, or

(ii) the splits corresponding to the edges in *A* and *B*, or

(iii) the quadruple topologies.

The distances, the split sets or the quadruple sets must be identical. For these three cases, the corresponding procedures have polynomial time complexity.

2.1. Score function, upper bound and elimination procedure

A quadruple is said to be *compatible* with an *X*-tree set, if all its topologies in the different *X*-trees are either non resolved or identical. Consequently, it is incompatible if it presents at least two different resolved topologies in two trees. Clearly, compatible quadruples can be assembled in a common tree structure.

Let the score function Sc: $X \rightarrow \mathbb{N}$ be defined as the number of quadruples containing *x* that are compatible with the trees, and *ScMax* be its maximum value over the *X* set of *n* elements:

$$Sc(x) \leq ScMax(n) = \frac{(n-1)(n-2)(n-3)}{6} = ScMax(n-1) \times \frac{n-1}{n-4}.$$

The first *ScMax* values are given in the following table:

n	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ScMax(n)	4	15	20	35	56	84	120	165	220	286	364	455	560	680	816	969

This allows to calculate an upper bound of the maximum number of elements admitting a common topology.

Proposition 1. The MAST size is lower than or equal to the highest value m such that $|\{x \text{ such that } Sc(x) \geq ScMax(m)\}| \geq m$.

Proof : There cannot exist a partial common tree with m leaves if there are less than m elements in X having a score larger than or equal to ScMax(m).

The elimination of the elements having a score lower than *ScMax(m)* is not a safe strategy even if they cannot belong to a common subtree with *m* leaves. Because, if finally the largest computed common tree has $m'<m$ leaves, some elements *x* having a score $ScMax(m') \leq Sc(x) < ScMax(m)$ would be eliminated. However, they can belong to a common subtree with $m'+1$ leaves or more. Nevertheless, we eliminate them for the moment, and we only deal with elements having a score larger than or equal to *ScMax(m)*. We denote X' the remaining elements, and set $n' = |X'|$.

2.2. One by one elimination

For each *x* in X', let *Nq(x)* be the number of incompatible quadruples containing *x* and *NbQuad* the whole number of incompatible quadruples on X'.

$$\sum_{x \in X'} Nq(x) = \sum_{x \in X'} ScMax(n') - Sc(x).$$

If $NbQuad>0$, at least one element must be deleted. This is a classical problem that is to cover a set with a minimum number of given subsets. The whole set contains all the incompatible quadruples, and

the n' subsets correspond to the incompatible quadruples containing one given element. Erasing one of them eliminates the corresponding subset, reduces the whole number of incompatible quadruples and set n' to $n'-1$.

This covering problem is well known to be NP-hard. The proposed method consists in deleting at each step one element covering the largest number of incompatible quadruples, that is the one having the largest Nq value. Clearly it is a greedy algorithm, since it never comes back on previous eliminations. When only compatible elements remain, a supplementary procedure is performed. It tries to reintroduce one by one the eliminated elements, in case their incompatibility was due to elements that have been erased later. Doing so, the selected set of taxa Y is maximal, since it cannot be extended.

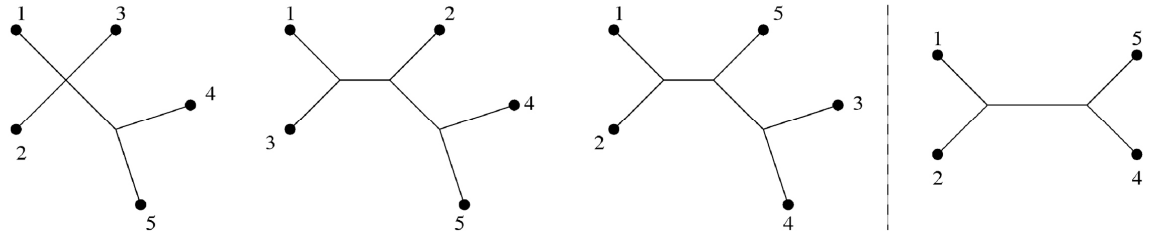

Figure 1. Three X-trees (left) and their largest common subtree (right); the Nq values are respectively equal to 3,3,4,3,3, leading to the elimination of leaf 3.

3. Algorithm

The X-trees are given in the *newick* format, which is poorly adapted to computation because it imposes an artificial root and, for a non resolved tree, it could contain arbitrary edges with null length. The first initial procedure consists in transforming each tree into two data structures: an unitary distance array and a table of all the splits corresponding to the edges with positive length. Notice that it is out of question to memorize the covering relation, since the usual values of n (≈ 100) makes such a task unrealistic.

3.1. The LAST Algorithm

```
/*Score computing*/
For all quadruple (x<y<z<t)
 if all the topologies are compatible
  Sc[x]++, Sc[y]++, Sc[z]++, Sc[t]++
End of For All

Determine the maximum number m of compatible elements.
Eliminate from X the elements x such that Sc[x]<ScMax[m]
Let Y be the remaining set

/*Recursive elimination*/
While (NbQuad>0)
 NbQuad:=0
 For all quadruple (x<y<z<t) of remaining elements
  If all the topologies are not compatible
   NbQuad++
   Nq[x]++, Nq[y]++, Nq[z]++, Nq[t]++
 End For all
 Eliminate one element with maximum Nq ;
End of While

/* Insertion of eliminated elements */
For all eliminated element x
 For all triples (y<z<t) of elements in Y
  If all the {x,y,z,t} topologies are compatible
   Y <- x
 End of For all
End of For all
```

Let Y be the final subset of X given by the LAST algorithm. It is clear that all the quadruples in Y are compatible and that Y is maximal in X. But nothing proves that Y is unique and has the MAST property. This can be partially tested by temporarily erasing an element in Y and looking for other compatible elements as in the insertion procedure. If this truncated Y can be extended with another element, an equivalent solution will be found and, if it can be extended again, a better solution could be detected.

3.2. Establishing the most resolved common tree

To build the common tree for which any node which is resolved at least once is resolved, starting from a single initial *X*-tree, involves too much edge processing. The simplest way is to add all the unitary distances restricted to *Y*, using a well known property of tree distances:

Proposition 2. Let A and B be two X-trees and D_A et D_B their associated tree distances (unitary or path length). The sum $D_A + D_B$ is a tree distance iff their topologies are compatible.

Thus, if a quadruple is only resolved once, it will necessarily be compatible and the most resolved common tree will appear. To get it from the sum of distances, any consistent algorithm can be applied; the optimal ones are in $O(n^2 \log n)$ for a tree distance, which is the case.

3.3. Complexity

The initial step establishing unitary tree distances is in $O(pn^3)$ and it is run just once. The score computation, including the elimination step of elements having a minimal score is in $O(pn^4)$ at each iteration. Their number being bounded by *n*, this heuristic is in $O(pn^5)$. The reintegration of the at most *n* elements is in $O(pn^4)$.

Nevertheless the program in C is fast, since it takes for instance, less than 10 seconds for two trees with 100 leaves, and 100 seconds for 100 trees with 30 leaves. It uses a limited memory space, pn^2 integer values for distances, and a few arrays with *n* or *2n* positions. It is freely available at http://www.bioinformatics.lif.univ-mrs.fr.

4. Significance of the common tree size

We have made many simulations (not reported here), with more than 2000 problems ($n<100$, $p<100$) with a given size of a MAST. Our algorithm always found the optimal value. So we use our program to compute statistics on the MAST size.

When, for a *(n,p)* problem, a largest common subtree size *m* is obtained, the next step to undertake is to know how far it is from the expected value under the null hypothesis, claiming that these *p* trees are *independent*. In order to answer this question, we generated, for each *(n,p)* values, 500 random sets of *X*-tree. Then, we computed the average of the values *m* and the critical value at 5%, denoted μ, such that the proportion of trees giving a value $m \geq \mu$ is not larger than 5%. Thus, each time the computed value *m(n,p)* is greater than or equal to μ*(n,p)* we reject the null hypothesis of independence to conclude that these trees have some similarities.

Let's comment the row of Table 1 corresponding to $n=50$. For $p=2$, both trees have on the average 11.58 compatible elements. This value results from the distribution detailed in Table 2 (rescaled in percentage). It thus appears that the critical value at 5% is equal to 15, since it is necessary to include the 6 cases giving $m=14$ to reach 95%.

Table 1. Average number of elements having the same topology in p random X-trees with n leaves. The values between parenthesis indicate the percentage of problems with a single conserved quadruple; the other problems have none. The critical value is shown after the pipe (|).

p	2	3	4	5	10	20
n=10	5.42 | 8	4.32 | 6	3.99 (99) | 5	3.35 (83) | 5	3.06 | 4	3 | 4
n=20	7.63 | 10	5.31 | 7	4.38 | 6	4.11 | 6	3.19 (19) | 5	3 | 4
n=30	9.12 | 12	5.97 | 8	4.70 | 7	4.23 | 6	2.58 (64) | 5	3 | 4
n=50	11.58 | 15	6.90 | 9	5.21 | 7	4.42 | 6	3.95 (95) | 5	3 | 4
n=75	13.85 | 17	7.68 | 10	5.64 | 8	4.67 | 7	3.92 (91) | 5	3 | 4

Table 2. Number of problems with 2 *X*-trees sharing a set of *m* compatible taxa.

m	9	10	11	12	13	14	15
Nb. Prob.	5	16	28	28	15	6	2

For $n=50$ and $p \geq 10$, there are less than 4 compatible elements. The value 3.95 corresponds to 95 trees that share a single compatible quadruple, and 5 trees having none; in this case, there are always at least 3 compatible elements, since no topology is required. For $p \geq 20$, an average value equal to 3 means that there's no conserved quadruple (4 vertices with a score equal to 1) and thus 4 is the critical value.

One can conclude that one compatible quadruple is sufficient to ensure that more than 20 phylogenetic trees share a common evolutive history part, whatever the number of taxa is.

References

Amir, A. and Keselman, D. 1997. Maximum agreement subtree in a set of evolutionary trees: Metrics and efficient algorithms, *SIAM Journal on Computing* 26: 758–769.

Barthélemy, J. P. and Guénoche, A. 1991. *Trees and Proximity Representations*. Wiley, New York.

Berry, V. and Nicolas, F. 2004. Maximum agreement and compatible supertrees, in *Proceedings of CPM*, 205–219.

Cole, R.; Farach, M.; Hariharan, R.; Przytycka, T. and Thorup, M. 1996. An O(n log n) algorithm for the maximum agreement subtree problem for binary trees, *SIAM Journal on Computing* 30: 1385–1404.

Robinson, D. F. and Foulds, L. R. 1981. Comparison of phylogenetic trees, *Mathematical Biosciences* 53: 131–147.

Semple, C. and Steel, M. 2003. *Phylogenetics*. Oxford University Press.

Steel, M. and Warnow, T. 1993. Kaikoura tree theorems: computing the maximum agreement subtree, in *Inf. Process. Lett.* 48(2): 77−82.