A Novel Quartet-Based Method for Phylogenetic Inference

B. B. Zhou, M. Tarawneh, C. Wang, A. Zomaya, School of I T, University of Sydney, NSW 2006, Australia
R. P. Brent, Mathematical Science Institute, Australian National University, Canberra, ACT 0200, Australia

1. Introduction

This paper introduces a novel quartet-based algorithm. The algorithm first calculates the likelihood values of the three possible resolved trees for each quartet and transforms them into three posterior probabilities (or quartet weights), and then it accumulates quartet weights to generate a global quartet-weight matrix. Using the topological information provided by the matrix, it recursively merges small sub-trees to larger ones until the final tree is obtained.

2. Global Quartet Weight Matrix

A quartet is associated with one of three fully resolved unrooted trees. (See Fig. 1.) The maximum-likelihood (ML) criterion can be used to measure which tree is more likely associated with a given quartet. The likelihood values for the three trees can be transformed into three posterior probabilities, or quartet weights [1].

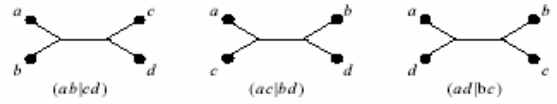

Fig. 1. The 3 possible fully resolved trees for a quartet {a, b, c, d}.

Let $(ij \mid kl, w)$ denote a quartet tree with weight w. With a complete set of $\binom{n}{4}$ quartets from n sequences, we can generate a symmetric matrix of size $n \times n$ by adding w's to the corresponding entries ij, ji, kl and lk.

3. Tree Topology and the Quartet Weight Matrix

There is a one-to-one mapping between a given tree topology and its associated quartet weight matrix. For mapping from tree topology to quartet weight matrix, the entry value m_{ij} in the matrix can be obtained by calculating the total number of quartet trees of the form $(ij \mid pq, 1.0)$, for p, or $q \neq i$, or j. Using the dynamic programming technique, we can obtain an $O(n^2)$ algorithm for generating the weight matrix according to a given tree topology.

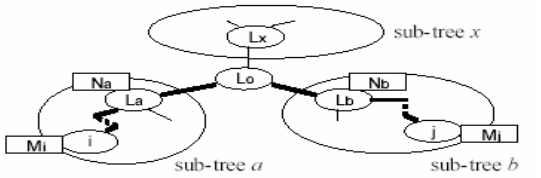

Fig.2. Each internal node is associated with three sub-trees a, b, and x in an unrooted tree. If we know n_a and n_b, the number of leaf nodes in sub-trees a and b; m_i, the number of concerned forms generated from leaf node i to leading node L_a in sub-tree a and m_j, the number generated from j to L_b in sub-tree b, we can obtain the entry value m_{ij} in the quartet matrix:

$$m_{ij} = \binom{n - (n_a + n_b)}{2} + m_i + m_j.$$

With a simple modification the algorithm for weight matrix generation we can obtain an algorithm for reconstruction of the tree topology from its generated matrix. The trick is that initially every row corresponds to a sub-tree with a single node and then we recursively calculate d_{ij} for a pair of sub-trees using the same equation in the above for m_{ij} and compare it with the corresponding entry value in the matrix using a confidence value $c_{ij} = m_{ij} / d_{ij}$. If c_{ij} is equal to one, the two sub-trees must have a common parent in the original tree. We need not to calculate d_{ij} for every pair of leaf nodes, one from each sub-tree. If the value for one pair of nodes is equal to the corresponding value in the matrix, all others will be the same.

This one-to-one mapping means that we can correctly reconstruct the evolutionary tree using the quartet weight matrix if all quartets from a given set of sequences are correctly resolved.

4. Tree Construction from Inaccurate Global Quartet Weight Matrices

It is very hard to have all the quartets fully and correctly resolved. Therefore, the matrix generated from a complete set of quartets for a given problem will be inaccurate, and the algorithm for tree topology reconstruction from its generated weight matrix cannot

be used without modification. To deal with inaccurate weight matrices we make three major changes to the original algorithm.

Average confidence value $\bar{c}_{ij}$: calculate the confidence values for all leaf node pairs, average them and this averaged value will be used as the confidence value $\bar{c}_{ij}$ for each pair of sub-trees.

Quartet weight correction: After two sub-trees are merged, we restore the associated entries in the matrix to their "true" values, i.e., change the quartet weights based on the currently reconstructed sub-trees and update the weight matrix accordingly. This is a very important procedure for us to change the direction of tree reconstruction.

Multiple tree reconstruction: Since the matrix is not accurate, it may not always be the right decision to merge the two sub-trees that have the highest confidence value. After the highest confidence value $\bar{c}_{ij}$ is obtained, we check whether there is another sub-tree k which has a reasonably high confidence value associated with one of the two sub-trees i and j, that is, we check whether $\bar{c}_{ik} \geq \alpha \bar{c}_{ij}$, or $\bar{c}_{jk} \geq \alpha \bar{c}_{ij}$ for α being a threshold which is smaller than, but close to one. At each of these critical points we can have three super quartet trees with four sub-trees i, j, k, and the rest as its four super nodes at different places. We keep all three different patterns. This will cause more trees to be generated. We use a parameter s to limit the total number of trees. Each time a critical point is encountered, two extra trees are generated until s such stages are encountered for each tree. Therefore, the maximum number of trees to be generated will be limited to 3^s.

5. Experimental Results

In the experiment we use the benchmarks consisting of 48,000 synthetic data sets of DNA sequences developed by the LIRMM Methods and Algorithms in Bioinformatics research group [2]. In our experiment we set α to 0.85, and s to 0, 4, 5, and ∞, respectively. We measure the average number of trees generated per data set and the percentage of correctly inferred trees and compare our results with the others obtained using currently popular methods, i.e., DNAPARS (the parsimony program), BIONJ (a neighbour joining method) and FASTDNAML (the maximum-likelihood program) for the same test data sets. Some experimental results are presented in Table 1. The results show that our algorithms performs much better than other methods in terms of correctly inferred trees, except our basic version (when s is set to 0) which construct only one tree for each data set.

		AA	BB	AB	CC	DD	CD
MD = 0.1 :	DNAPARS	19	16	15	13	15	17
	BIONJ	17	16	15	16	15	18
	FASTDNAML	23	20	21	16	18	18
	QBNJ (s=0)	22	11	9	10	10	10
	QBNJ (s=4)	63/57	49/53	52/54	56/45	61/47	59/44
	QBNJ (s=5)	68/94	57/89	58/82	58/68	61/69	61/65
	QBNJ (s= ∞)	70/124	58/117	59/120	58/86	64/88	62/80
MD = 0.3 :	DNAPARS	28	36	26	46	53	51
	BIONJ	33	34	34	56	57	56
	FASTDNAML	58	53	54	70	68	69
	QBNJ (s=0)	44	30	21	37	32	32
	QBNJ (s=4)	87/44	78/40	79/46	88/24	91/26	89/26
	QBNJ (s=5)	91/61	81/55	83/60	89/28	92/28	90/31
	QBNJ (s= ∞)	92/66	82/61	83/68	89/29	92/28	90/32
MD = 1.0 :	DNAPARS	6	8	6	7	9	10
	BIONJ	22	20	21	62	63	65
	FASTDNAML	44	36	37	82	83	81
	QBNJ (s=0)	43	19	13	41	39	32
	QBNJ (s=4)	86/54	67/51	65/56	88/31	90/31	90/32
	QBNJ (s=5)	91/79	72/79	72/87	89/38	94/37	92/39
	QBNJ (s= ∞)	91/89	74/90	72/101	90/41	95/37	92/42
MD = 2.0 :	DNAPARS	0	0	0	0	0	0
	BIONJ	2	3	3	29	31	33
	FASTDNAML	8	4	5	59	62	59
	QBNJ (s=0)	12	2	2	9	10	13
	QBNJ (s=4)	45/71	19/73	20/72	45/59	65/57	58/59
	QBNJ (s=5)	56/144	26/158	29/167	53/100	70/94	65/100
	QBNJ (s= ∞)	60/207	31/344	34/294	55/129	71/124	67/130

Table 1. Some experimental results. The figure x/y denotes the percentage of correctly inferred trees / the average number of trees generated per data set.

6. Conclusions and Future work

The experimental results show that the probability is very high for the correct tree to be among a very limited number of trees constructed using our method. We can also provide important information on where the trees differ (critical points).

One may argue that it is hard to believe other popular method, such as the maximum likelihood, are unable to achieve similar, or even better results than those obtained using our method if multiple trees are allowed to be generated. However, the comparison of simulation results among different methods presented in this paper is fair: If the maximum likelihood is able to correctly identify the correct trees, it is clear that just adding one more procedure using the ML at the end, our method can significantly outperform most existing popular methods on the basis of constructing only a single tree!

If we can find good optimisation criteria which enable us to choose the right direction directly at each critical point, we shall obtain a method which can outperform the existing popular methods, without the need to generate multiple trees for a given problem.

Our method is quartet based and it is important to find more efficient methods to improve the quality of quartet trees, or reduce quartet errors.

10. References

[1] K. Strimmer, and A. von Haeseler, (1997) Likelihood–mapping: A simple method to visualize phylogenetic content of a sequence alignment, *Proc. Natl. Acad. Sci.* USA, 94, 1997, pp.6815–6819.

[2] The LIRMM Methods and Algorithms in Bioinformatics research group, www.lirmm.fr.