A New Algorithm for Reconstruction of Phylogenetic Tree*

ZhiHua Du and Zhen Ji

ShenZhen Unvierstiy, Shenzhen,China
du_zhihua@yahoo.com.cn

Abstract. The abstract should summarize the contents of the paper and should contain at least 70 and at most 150 words. It should be set in 9-point font size and should be inset 1.0 cm from the right and left margins. There should be two blank (10-point) lines before and after the abstract. This document is in the required format. In this paper, we present a new algorithm for reconstructing large phylogenetic tree. This algorithm is based on a family of Disk-Covering Methods (DCMs) which are divide-and-conquer techniques by dividing input dataset into smaller overlapping subset, constructing phylogenetic trees separately using some base methods and merging these subtrees into a single one. Provided the high memory efficiency of RAxML (which the program inherited from fastDNAml) compared to other programs and the good performance on largereal-world data it appears to be best-suited for use as the base method. The experiments clearly show that the proposed algorithm improves over stand-alone RAxML on all datasets, i.e. yields better likelihood values than RAxML in the same amount of time. This results serve as an argument for the choice of the proposed algorithm instead of stand-alone RAxML.

Keywords: Phylogenetic tree,divide-and-conquer, DCM.

1 Introduction

Phylogenetic tree illustrates the evolutionary relationships among a groups of organisms, or among a family of related nucleic acid or protein sequences, e.g., how might have this family been derived during evolution. It plays a fundamental role in many biological problems such as multiple sequence alignment, protein structure and function prediction, and drug design [1].

There are two general categories of methods for calculating phylogenetic trees: distance-based and character-based. Distance-based methods compute a matrix of pairwise distances between sequences in an alignment, and then constructing a tree based entirely on the original distance computations instead of sequences. There exist many methods based on this idea. Such as, Neighbor-Joining [2] and other improved

* This work is partially funded by a Research Foundation granted by the Shenzhen University under grant no: 4DZH), the National Natural Science Foundation of China under grant no. 60572100, the Royal Society (U.K.) International Joint Projects 2006/R3 - Cost Share with NSFC (China).

H. Li et al. (Eds.): AIRS 2008, LNCS 4993, pp. 472–477, 2008.

distance methods, WEIGHBOR [3], BIONJ [4], FASTME [5] and a latest approach considering maximum-likelihood estimated triplets of sequences [6]. Disadvantages of distance-based methods include the inevitable loss of evolutionary information when a sequence alignment is converted to pairwise alignment [9].

Character-based methods would examine each column of the alignment separately and look for the tree that best accommodates all of this information, such as maximum parsimony (MP) [7] or maximum likelihood (ML) [8]. MP chooses tree that minimizes number of changes required to explain data. ML, under a model of sequence evolution, finds the tree which gives the highest likelihood of the observed data. Character-based methods are information rich for there is a hypothesis for every column in the alignment. However, The MP method is NP-hard, and ML has unknown complexity [10] and is very hard to solve in practice. Primary sources of phylogenetic tree construction software include the PHYLIP website (http:// evolution. genetics.washington.edu/phylip.html), MrBayes [11] and PAUP [12][13]

Previous studies have shown that large datasets are challenging for MP heuristics implemented in these packages [14][15]. To analyze datasets containing thousands of sequences Disk Covering Methods (DCMs)[14][16][17][18] were introduced. To the best our knowledge, Recursive-Iterative-DCM3 (Rec-I-DCM3), is the best known technique heuristic for solving MP.

AS to ML, one of the recent methods, RAxML [19][20], is among the fastest, most accurate, and most memory-effcient ML heuristics on real biological datasets to the best of our knowledge. Furthermore, the global optimization method (fast Nearest Neighbor Interchange adapted from PHYML [21] is not as efficient on real alignment data as RAxML. Thus, it is not suited to handle large real-data alignments of more than 1,000 sequences.

In this paper, we present a new algorithm for reconstructing large ML phylogenetic tree by integrating algorithmic concepts from Rec-I-DCM3 and RAxML. The experimental results show that the proposed algorithm outperforms the existing methods.

2 Methods

The proposed algorithm is consists of four main steps.

Step 1: Recursively divide the given dataset into smaller, overlapping subproblems, until the subproblems become at most of size Maximum subproblem size (MS) and stores the merging order (subset-guidetree, rurTree) which is required to correctly execute the merging step.

Step 2: Construct phylogenetic tree for subproblems by using the RAxML method.

Step 3: Combine the sub phylogenetic trees into a unique supertree ´ ´

Step 4: A hill-climbing ML search on the supertree, T', is applied to do a global rearrangement

Some definitions used are shown as following:

Short subtree. Suppose there is a tree T with an edge e in it. Let Q_1, Q_2, Q_3 and Q_4 be the four subtrees around e; q_1, q_2, q_3 and q_4 be the set of leaves closet to e in each

of the four subtrees respectively. The distance between them is measured by the hamming distances on the edges. The set of nodes in $q_1 \cup q_2 \cup q_3 \cup q_4$ is the short subtree around e.

Short subtree graph. Short subtree graph is the union of cliques formed on "short subtrees" around each edge in T

Separator. Separator is the short subtree of a special edge, which would produce the most balanced bipartition of the leaves in tree T when removed.

The outline of the proposed algorithm is as following:

1. Problem Initialization
   1.1 Set S={S0, …, Sk-1} of aligned biomolecular sequences. Set k=number of sequences, n=number of iteration, b=base heuristic (TNT), T=starting tree, MS=maximum subproblem size.
   1.2 Initialize a subset guide-tree, rurTree, to record recursive calls as the topology for merge subtrees.
   Initialize, allsubsets, to save a total set of subproblems.
2. For n iterations do
   2.1 /*Construct a recursive DCM3 decomposition using T|S (a guidetree tree on dataset S) as the guide tree, producing a total set of subproblems, allsubsets =A0, A1 ,…, Am-1 (m is the total number of subsets). Produce a subset guide-tree, rurTree, to keep the merge order. The rurTree is expressed in a string format that uses parenthesis to start and end subtree groups, commas to separate group members, and subproblems names to name tree leaves. */
   Recursive_Divide(S, MS, T)
   2.2 Build phylogenetic subtrees of subproblems by using RAxML.
   2.3 Use a postorder tree walk algorithm to search subset-guidetree, rurTree, in order to merge subtrees into a suptertree, T'.
   2.4 Apply hill-climbing heuristic starting from T' until we reach a local optimum. Let T' be the resulting local optimum. Set T=T'.

Function Recursive_Divide(S, MS, T)

Input: Set of k sequences S = S1; S2; :::; Sk Maximum subset size MS
Starting tree T
Output: Set of subproblems, allsubsets = A1,A2, …,Am
(m is the total number of subsets)
Algorithm: Recursively divide a set of k sequences S into subproblems

(a) Compute edge weighting for each edge by using the hamming distances.
(b) Compose short subtree graph around edges by selecting set of all leaves that are elements in a short quartet around an edge, that is $sub_1, sub_2, \ldots, sub_x$ (where x is the number of subsets).
(c) Find a separator, spt, by selecting an edge that when removed, produces the most balanced bipartition of the leaves as centroid edge, Ec.

The spt is the leaves of the short subtree around Ec. The subsets are then defined to be

$$A_i = spt \cup sub_i, 1 \le i \le x$$

(d) For $A_i (1 \le i \le x)$
If (A_i 'size >MS){
Let T|A_i be the result of restricting tree T to Ai for each i.
/*Recursively compute the subsets for A_i */
Recursive_Divide (A_i, MS, T| A_i)
}
Else{
Add A_i to allsubsets.
Re-build subset-guidetree, rurTree.
/*Produce a subset-guidetree, rurTree, to keep the merge order. The rurTree is expressed in a string format that uses parenthesis to start and end subtree groups, commas to separate group members, and subproblems names to name tree leaves.*/
}

3 Experimental Design

The online version of the volume will be available in LNCS Online. Members of institutes subscribing to the Lecture Notes in Computer Science series have access to all the pdfs of all the online publications. Non-subscribers can only read as far as the abstracts. If they try to go beyond this point, they are automatically asked, whether they would like to order the pdf, and are given instructions as to how to do so.

Methodology

To the best our knowledge, RAxML, is the best known technique heuristic for solving ML. Therefore, we design an experiment to show that the proposed algorithm would improve over stand-alone RAxML on all datasets in order to demonstrate the benefits which arise from using the dividing method.

D**atasets**

The experiments were done on six large datasets, some of which are obtained from http://www.cs.njit.edu/usman/RecIDCM3.html. The datasets we used are (1) 6281 Eukaryotes ssu rRNA sequences from the European rRNA database, (1661 sites), (2) 6458 firmicutes bacteria 16s rRNA sequences from the RDP (1352 sites), (3) 6722 three-domain rRNA sequences from Robin Gutell (1122 sites) [22], (4) 7769 three-domain + 2 organelle rRNA sequences from Robin Gutell (851 sites), (5) 11361 set of all bacteria ssu rRNA sequences from the European rRNA database (1360 sites)[23], and (6) 13921 proteobacteria 16s rRNA sequences from the RDP (1359 sites) [22].

4 Experimental Results

In our experiments both methods start optimizations on the same starting tree. Due to the relatively long execution time on large alignments we only executed one iteration

per dataset. The run time of one the proposed algorithm iteration was then used as inference time limit for RAxML. Table 1 provides the log likelihood values for RAxML and the proposed algorithm after the same amount of execution time. Note that, the apparently small differences in final likelihood values are significant because those are logarithmic values and due to the requirements for high score accuracy in phylogenetics (T.L.Williams et al. 2004).

Table 1. The proposed algorithm versus RAxML log likelihood values after the same amount of inference time

Dataset	Proposed algorithm	RAxML
Dataset1	-99967	-99982
Dataset2	-355071	-355342
Dataset3	-383578	-383988
Dataset4	-1270920	-1271756
Dataset5	-901904	-902458
Dataset6	-541255	-541438

The experiments clearly show that the proposed algorithm improves over stand alone RAxML on all datasets, i.e. yields better likelihood values than RAxML in the same amount of time. This results serve as an argument for the choice of the divide-and-conquer method instead of stand-alone RAxML.

References

1. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)
2. Bull, J.J., Wichman, H.A.: Applied evolution. Annual Review of Ecology and systematics 32, 183–217 (2001)
3. Saitou, N., Nei, M.: The nigehbor-joining method: a new method for reconstructing phylogenetic tree. J Mol Evol 4, 406–425 (1987)
4. Bruno, W.J., Socci, N.D., Halpern, A.L.: Weighted Neighbor Joining: A Likelihood-Based Approach to Distance-Based Phylogeny Reconstruction. Mol Biol Evol 17, 189–197 (2000)
5. Gascuel, O.: BIONJ: an improved version of the NJ algorithm based on a simplemodel of sequence data. Mol Biol Evol 14, 685–695 (1997)
6. Desper, R., Gascuel, O.: Fast and Accurate Phylogeny Reconstruction Algorithms based on the Minimum-Evolution Principle. J Comput Biol 19, 687–705 (2002)
7. Ranwez, V., Gascuel, O.: Improvement of Distance-Based phylogenetic Methods by a Local Maximum Likelihood Approach Using Triplets. Mol Biol Evol 19, 1952–1963 (2002)
8. Camin, J., Sokal, R.: A method for deducing branching sequences in phylogeny. Evolution 19, 311–326 (1965)
9. Felsentein, J.: Evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol 17, 368–376 (1981)
10. Steel, M.A., et al.: Loss of information in genetic distances. Nature 336, 118 (1988)
11. Stelel, M.A.: The maximum likelihood point for a phylogenetic tree is not unique. Systematic Biology 43(4), 560–564 (1994)

12. Huelsenbeck, J.P., Ronquist, F.: MYBAYES: Bayesian inference of phylogenetic trees. Bioinformatics 17, 754–755 (2001)
13. Swofford, D.: PAUP*. Phylogenetic Analysis Using Parsimony (* and other mothods). Version 4.Sinauer Associates (2002)
14. Sjolander, K.: Phylogeneomi inference of protein molecular function: advances and challenges. Bioinformatics 20, 170–179 (2004)
15. Roshan, U.: Algorithm techniques for improving the speed and accuracy of phylogenetic methods. PhD thesis (2004)
16. Roshan, U., Moret, B.M.E., Warnow, T., Williams, T.L.: Rec-i-dcm3: a fast algorithmic technique for reconstructing large phylogenetic trees. In: Proceedings of the IEEE Computational Systems Bioinformatics conference (CSB), Stanford, California, USA (2004)
17. Huson, D., Nettles, S., Warnow, T.: Disk-covering, a fast-converging method for phylogenetic tree reconstruction. Journal of Computational Biology 6, 369–386 (1999)
18. Nakhleh, L., Roshan, U., St. John, K., Sun, J., Warnow, T.: Designing fast converging phylogenetic methods. In: Proc. 9th Int'l Conf. on Intelligent Systems for Molecular Biology (ISMB 2001). Bioinformatics, vol. 17, pp. S190–S198. Oxford U. Press, Oxford (2001a)
19. Warnow, T., Moret, B., St. John, K.: Absolute convergence: True trees from short sequences. In: Proc. 12th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA 2001), pp. 186–195. SIAM Press, Philadelphia (2001)
20. Stamatakis, A., Ludwig, T., Meier, H.: Parallel inference of a 10.000-taxon phylogeny with maximum likelihood. In: Proceedings of 10th International EuroPar Conference, pp. 997–1004 (2004)
21. Stamatakis, A., Ludwig, T., Meier, H.: Raxml-iii:a fast program for maximum likelihood-based inference of large phylogenetic trees. Bioinformatics 21(4), 456–463 (2005)
22. Guindon, S., Gascuel, O.: A simple, fast, and accurate algorithm to estimatelarge phylogenies by maximum likelihood. Syst. Biol. 52(5), 696–704 (2003)
23. Maidak, B., et al.: The RDP (ribosomal database project) continues. Nucleic Acids Research 28, 173–174 (2000)
24. Wuyts, J., Van de Peer, Y., Winkelmans, T., De Watchter, R.: The European database on small subunit ribosomal RNA. Nucleic Acid Research 30, 183–185 (2002)