Dynamic and Parallel Construction of Evolutionary Tree using Clustering

Tanjil Ahmed, A. K. M. Saifun Nabi,

Zalia Shams, Kazi Zakia Sultana, Anupam Bhattacharjee

Department of Computer Science and Engineering.
Bangladesh University of Engineering and Technology,
BUET, Dhaka-1000, Bangladesh.
tanim_buet62@yahoo.com, shabuj103@yahoo.com,
setu_18@yahoo.com, z_sultana00@yahoo.com, abrbuet@yahoo.com

Abstract

Phylogenetic trees are commonly constructed based on hard optimization problems such as Maximum Parsimony (MP), Maximum Likelihood (ML) and ZAZ method. Conventional MP heuristics for producing phylogenetic trees produce good solutions within reasonable time on small database while ML heuristics are limited to smaller datasets. Dynamic ZAZ approach deals with relatively large database but it has some limitations. In this paper, we introduce an optimal clustering algorithm, a popular data mining technique, for the dynamic and parallel construction of evolutionary trees using CREW-PRAM model. The algorithm has been implemented and tested against large biological datasets ranging from 5,000 to 7,000 sequences and dramatic speedup with significant improvement in accuracy in comparison to existing approaches has been observed.

Keywords: Bioinformatics, Clustering, Data Mining, Evolutionary Tree Construction, Parallel Algorithm, Shared Memory Model.

I. INTRODUCTION

An *evolutionary tree* (*phylogenetic tree*) is a graphical representation of phylogenetic analysis that represents the evolutionary relationships among groups of organisms or among a family of related nucleic acids or protein structures, i.e., how this family might have been derived during evolution. Leaves of the tree represent genes or species and internal nodes of the tree are hypothetical ancestral units [1], [13].

Construction of evolutionary tree from sequence data is a well-studied problem in Bioinformatics. Different heuristics are used as the problem is *NP*-Hard in general [6]-[9]. The sequences under consideration should be universally present in all organisms to be studied with good conservation of sequences among many of the species and at the same time, the sequences should be divergent enough to allow grouping the species into a taxonomy classification. Hence the accurate reconstruction of evolutionary tree is an interesting problem in computational biology [1], [9].

The information in a molecular sequence alignment can be used to compute a phylogenetic tree for a particular family of gene sequences. The branching in phylogenetic trees represents evolutionary distance based on sequence similarity scores or on information-theoretic modeling of the number of mutational steps required to change one sequence into the other.

A well-known variant of the problem of constructing an evolutionary tree is based on experiments. An experiment determines how three species are related in the evolutionary tree i.e. returns the topological subtree (without weights on the edges) for the three species. Fig. 1 shows such an example of a tree and the outcome of experiments. The approach involves generating a topology on the taxa using a model [14], then generating a sequence label for the root and finally evolving the sequence along the paths in the tree to obtain a label for each leaf [2], [3], [15]. The efficiency and performance of the algorithm is judged based on how closely the inferred tree matches the target tree. Several studies have been performed on the supertree methods, relying almost exclusively on simulated data [4], [5].

All the methods described earlier are static in the sense that all sequences are available from the beginning of the construction. So, when a new species appears the evolutionary tree needs to be rebuilt. There comes the concept of dynamic construction of evolutionary tree [10], [12]. Paper [16] has worked on that very recent concept. Again, parallel approaches of super tree construction have been studied in [11]. In the paper of Du, Roshan, and Lin [11] they present algorithm using the Multiple Instruction Multiple Data (MIMD) parallel computing method and applying recursive DandC approach with DCM3 method. On the other hand, in paper [16] the authors have studied on the new parallel approach dealing with CREW-PRAM parallel computing model and analyzed the load division phase. Combination of both the approaches (Dynamic and Parallel) have been studied in [16]. But the earlier works face some limitations (e.g. the equal length of all the sequences, improper similarity measures and lack of data mining techniques in case of large data sets). In this paper, we present an optimal clustering algorithm, a popular data mining technique, for the dynamic and parallel construction of evolutionary trees using CREW-PRAM model.

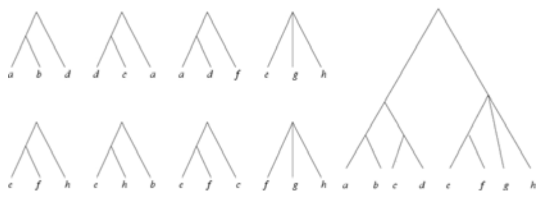

Figure 1: An example of experiment results and the corresponding evolutionary tree.

II. PRELIMINARIES

A. Shared Memory Model

The *Shared Memory Model* consists of a number of processors, each having its own local memory and executing its own local program by communicating through a shared memory unit. Each processor is uniquely identified by an index, called processor number or processor id. Of the two basic modes of shared memory model, in the synchronous (parallel random access machine, PRAM) mode, all the processors operate synchronously under the control of a common clock. Among several variations of PRAM model based on the simultaneous access of multiple processors to the same location of the global memory, the concurrent read exclusive write (CREW) PRAM model allows simultaneous access for a read instruction only.

B. Clustering

Clustering is the process of organizing data into several groups. But, unlike classification, here the groups are not predefined. Instead, the grouping is accomplished by finding similarities between data according to characteristics found in actual data. These groups are called clusters. So cluster is a collection of elements such that each element is as close as possible to some other member of the cluster.

Clustering algorithm may be viewed as *hierarchical* or *partitional*. A nested set of clusters is created with hierarchical clustering. On the other hand, with partitional clustering, the algorithm creates only one set of clusters. Traditional clustering algorithms deal with small numeric database that fit into memory. But more recent clustering algorithms deal with larger, perhaps dynamic databases. The clustering algorithm [17] used in this paper implements this approach. Clustering algorithms may also differ based on whether they produce overlapping or non-overlapping clusters. Even though we consider only non-overlapping clusters it is possible to place an item in multiple clusters. In turn, non-overlapping clusters can be viewed as extrinsic or intrinsic. Extrinsic technique implies traditional classification supervised learning algorithm in which a special input training is used. Intrinsic algorithms do not use any a priori category labels, but depend only on similarity measure. The algorithm for clustering used in the paper [17] falls into the intrinsic class. Especially this algorithm is based on CURE (Clustering Using REpresentatives) approach where representative genes are identified by approximated weighted length technique. These genes represent the clusters.

III. DYNAMIC CONSTRUCTION OF EVOLUTIONARY TREE

In static construction, when a new sequence arrives, the whole tree needs to be constructed again from the scratch even if the best heuristic (*ML*, *MP* etc.) is applied. The computational cost of this static method increases in proportion with the number of species. On the other hand, here, in dynamic approach, when a new sequence arrives, the tree is updated with this new sequence in such a way that it is very close to the static tree. Here we present an algorithm that can be used in dynamic construction of phylogenetic trees.

A. The Algorithm

Definitions:

n_i= Frequency of i type element in input database , $i \in \{A, C, T, G\}$

N= Total occurrence of each type of element= $\sum n_i$

w_i= Weight of i type element of gene sequence =n_i/N

W_g= Weight of the gene sequence g =$\sum w_i$

l_g= Length of the gene sequence g

ref= Reference gene of j-th subtree

W_{ref_j} = Weight of the reference gene of j-th subtree

l_{ref_j} = Length of the reference gene of j-th subtree

$S(g, ref_j)$ = Function denoting similarity measure between genes g and ref_j=$|W_{ref_j} \times l_{ref_j} - W_g \times l_g|$

min_dsim = min $S(g, ref_j)$ for all j-th subtree

$H(j)$=Threshold value of subtree j=$W_{ref_j} \times l_{ref_j} \times 55\%$

$\pounds$ = A linked list containing pointers to all isolated nodes and ancestor nodes.

T= The tree where

- Each node contains the sequence.
- The ancestor node contains $\sum_i W_i l_i$ and $\sum_i l_i$

A_T = Currently assigned subtree number

$Select\_Seq(A, B)$ = generates a sequence with maximum matching and mismatches replaced by dominating bases.

Input: String of sequences from *EMBL* database
Output: $T^{'}$ = The updated tree.

```
Algorithm Dynamic-STZAZ
Begin
1. Initialize $min\_dsim$ to INFINITY and $A_T$ to 0
2. Assign weight $w_i$ to each different element $i$ of gene sequence in input database.
3. Make the first sequence as the only node in the first subtree and mark it as reference gene of that tree
```

```
4. Calculate $H(1)$ according to definition
5. When a sequence $g$ arrives
5.1. For each node $i$ in $\mathcal{L}$
5.1.1.   Find $S(g, \text{ref}_i)$
5.1.2. if $S(g, \text{ref}_i)<H(i)$ and $S(g, \text{ref}_i) < min\_dsim$
5.1.2.1. $min\_dsim = S(g, \text{ref}_i)$
5.1.2.2. $A_T= i$
5.2. Calculate the weight of the sequence $g$
5.2.1. if it is a new sequence $g$
5.2.1.1. $W_g = \sum_k w_k \times f_k$, $f_k$ is the frequency of $k$-th element of the sequence $g$
5.2.1.2. $l_g = \sum_j f_j$, $f_j$ is the frequency of $j$-th element of the sequence $g$
5.2.2. else it is an ancestor node $g$
5.2.2.1. $W_g = \frac{W_j \times l_j}{\sum l_j}$, $j$ is the node of the tree
5.3.1. if $A_T \neq 0$
5.3.1.1. Update the tree by combining the new node with the old root and change the root. Let the new node and old node be $A$ and $B$ respectively.
         $g=Select\_seq(A,B)$.
5.3.1.2. $W_{avg} = \frac{W_{A_T} \times l_{A_T} + W_g \times l_g}{l_{A_T} + l_g}$
5.3.1.3. $l_g = \sum_j f_j$
5.3.1.4. Calculate the threshold for updated tree.
5.3.1.5. Search the updated tree to find out gene whose weight is closest to $W_{avg}$ and make it the new reference node.
5.3.1.6. goto step 5 with the ancestor node as $g$.
5.3.2. else if $g$ is a new sequence
5.3.2.1. Create a new entry in list $\mathcal{L}$ where reference gene is the new sequence.
5.3.2.2. $l_g = \sum_j f_j$
5.3.2.3. $W_g = \sum_j w_j \times f_j$
5.3.2.4.Calculate threshold value for gene sequence $g$
5.3.2.5  goto step 5
6. When no more sequence arrives
6.1. while $\mathcal{L}$ has more than one ancestor node
6.1.1. Decrease threshold of each node in $\mathcal{L}$ by 10%. Find out the two most similar ancestors $A_1$ and $A_2$ where $S(A_1, A_2)$ is minimum and
$S(A_1, A_2) < \frac{H(A_1) \times l_{A_1} + H(A_2) \times l_{A_2}}{l_{A_1} + l_{A_2}}$
In case of ties, take the two nodes with the highest threshold. Now $A_T= A_1$ and $g = A_2$.
6.1.2. goto 5.2
END
```

B. Description

At first, *min_dsim* is initialized to infinity, A_T to zero (step 1). Then weight w_i for each different element of gene sequence in the database is calculated as described in definition section (step 2).

When the first sequence arrives, it is inserted into list $\mathcal{L}$ and marked as the reference gene of that node (step 3). Then $H(1)$ is calculated (step 4). Now, when sequence g arrives, the list $\mathcal{L}$ is searched to find an element i of $\mathcal{L}$ for which dissimilarity is less than some predefined value such as *min_dsim* and threshold. Thus appropriate subtree is found (step 5.1). Then weight of the sequence g is calculated (step 5.2). Now 2 cases may arise.

Case 1 (step 5.3.1): g is assigned to a subtree of $\mathcal{L}$, the subtree is updated and new W_{avg} and new threshold is calculated. Then, the reference gene is found out for the updated tree. Now, g is treated as new ancestor node and step 5.1 is repeated.

Case 2 (step 5.3.2): Here g is a new sequence and not assigned to any existing subtree. A new entry is made in $\mathcal{L}$ for g, it is marked as reference gene and l_g, W_g and threshold are calculated. Then step 5.1 is repeated.

When no more sequences arrive, if $\mathcal{L}$ has more than one element threshold of each node is decreased by 10%. Then two most similar nodes are searched from $\mathcal{L}$ (step 7.1.1). In case of ties, the two nodes with highest threshold are selected. This process is repeated until $\mathcal{L}$ contains only one node, i.e., the tree T.

IV. DYNAMIC CONSTRUCTION OF EVOLUTIONARY TREES USING CLUSTERING

In the phylogenetic tree construction algorithm reference gene sequence is used which represents a cluster. Here we present a clustering algorithm for the reference gene calculation and optimal clustering. When a new gene sequence arrives, clustering or classifications are performed based on this reference sequence. Then using the clusters the dynamic construction algorithm is applied.

The work in the paper [17] proves that the total clustering approach is optimal as it generates same load on each processor. And again, if all the gene sequence comes with same length and if mismatch between the genes is not a factor of parallel processing, the optimal parallel program would generate the same load. For n sequence, the total running time of the algorithm is $O(n^2)$.

Observation:

By implementing the algorithm, we find that the number of clusters increases proportionately with the increase of the number of sequences. After a certain number of sequences, number of clusters remains almost static as similarity among the sequences increases at that stage. Fig. 2 shows the relationship between the number of clusters and the number of sequences. Here, the curves are obtained by varying the values of the threshold value from 0.2 to 1. The marked curve is for the average value of threshold = 0.55. The cross-marked line denotes unacceptable value of threshold 0.1.

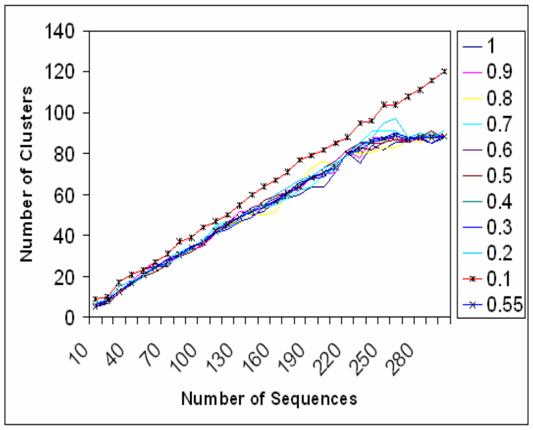

Figure 2: Graph obtained for different threshold values

V. PARALLEL ALGORITHM FOR PHYLOGENETIC TREE CONSTRUCTION

In this section, we present a generic parallel algorithm for phylogenetic tree construction.
According to [16] we find,

$$T(m, m_c) \geq \frac{ml^2(l+2)^{m_c}[m_c \lg(l+2) + 2\lg l]}{\lg p}$$

Now, to find the extreme value of m_c we differentiate the inequality with respect to m_c.

$$ml^2(l+2)^{m_c}\lg(l+2)[1+m_c \lg(l+2)+2\lg l] - \lg p \leq 0$$

To find the value of m_c we apply bisection method ranging from $m_c = 1 \sim n$. Thus we get the optimum value of m_c in $O(\lg n)$ iteration at most.

Algorithm parallel- STZAZ

```
BEGIN
    1. Divide loads based on $m_c$
    2. Apply dynamic-STZAZ parallely
    3. Construct the final tree
END
```

A. DESCRIPTION

The whole set of gene sequences is divided into clusters based on value of m_c found by solving the inequality. Dynamic-STZAZ is then applied parallely on these clusters. Thus, the complete tree is constructed.

B. ACCURACY

We calculate the accuracy of the algorithm by comparing the height of the experimentally found tree with that of the optimal tree. The height of the optimal tree with n leaves is $\lceil \lg n \rceil$. From Fig. 3, it is obvious that the accuracy of dynamic STZAZ is better than that of parallel STZAZ. Though parallel STZAZ is faster than dynamic STZAZ when the number of sequences is above 600, the accuracy of both parallel and dynamic STZAZ is almost same. The algorithm performs 10% better in accuracy than *ML* approach for small size of dataset (no. of sequence < 1000).

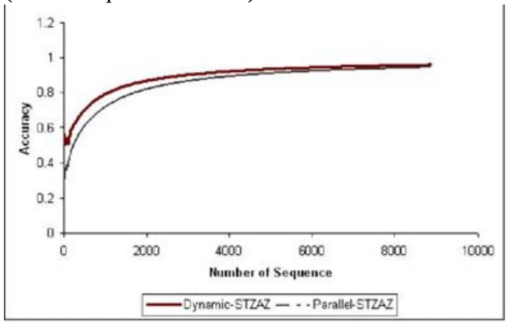

Figure 3: Graph obtained from experiment for two approaches

4. CONCLUSION

In this paper we presented evolutionary tree construction by applying clustering and parallel algorithms and dynamic construction. The new approach establishes a modified construction with much improvement in accuracy and speed.

REFERENCES

[1] Rastogi, Mendiratta, and Rastogi, "*Bioinformatics Methods and Applications. Genomics, Proteomics and Drug Discovery*," Prentics-Hall, New Delhi, 2004.

[2] Felsenstein, J., "Evolutionary trees from DNA sequences: a maximum likelihood approach," *J. Mol. Evol.*, 17, 368–376, 1981.

[3] Guindon, S., and Gascuel, O., "A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood," *Syst. Biol.*, 52, 696–704, 2003.

[4] Kuhner, M. K., and Felsenstein, J., "A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates," *Mol. Biol. Evol.*, 11, 459–468, 1994.

[5] Ranwez, V., and Gascuel, O., "Improvement of distance-based phylogenetic methods by a local maximum likelihood approach using triplets," *Mol. Biol. Evol.*, 19, 1952–1963, 2002.

[6] Rogers, J. S., and Swofford D. L., "A fast method for approximating maximum likelihoods of phy-

logenetic trees from nucleotide sequences," *Syst. Biol.*, 47, 77–89, 1998.

[7] Schmidt, H. A., Strimmer, K., Vingron. M. and Von Haeseler, "TREE-PUZZLE: maximum likelihood phylogenetic analysis using quartets and parallel computing," *Bioinformatics*, 18, 502–504, 2002.

[8] Foulds L., and Graham R., "The Steiner problem in phylogeny is *NP*-complete," *Advances in Applied Mathematics*, 3:43–49, 1982.

[9] Miller W., "Comparison of genomic sequences: solved and unsolved problems," *Bioinformatics*, 17:391-397, 2000.

[10] Ceron, C., J. Dopazo, E. L. Zapata, J. M. Carazo, and O. Trelles., "Parallel implementation of DNAml program on message-passing architectures," *Parallel Computing*, 24: 701-716, 1998.

[11] Zhihua D., Feng L., and Roshan U. W., "Reconstruction of Large Phylogenetic Trees: A Parallel Approach," *Electronic Version* (E-book), 2005.

[12] Roshan U., Moret B., Williams T., and Warnow T., "Rec-I-DCM3: A Fast Algorithmic Technique for Reconstructing Large Phylogenetic Trees," *Proceedings of the IEEE Computational Systems Bioinformatcis Conference*, 2004.

[13] Jones N., and Pevzner P., "*An Introduction to Bioinformatics Algorithms*," MIT Press, Cambridge, Massachusetts, London, England, 2004.

[14] Fitch W., "Toward defining the course of evolution: minimum change for a specified tree topology," *Syst. Zool.*, 2002.

[15] Felsenstein, J., "PHYLIP-Phylogeny Inference Package (Version 3.2)," *Cladistics*, 5, 164–166, 1989.

[16] Bhattacharjee A., Sultana K. Z., and Shams Z., "Dynamic and Parallel approaches to optimal evolutionary tree construction," *Canadian Conference on Electrical and Computer Engineering*, Ottawa Congress Centre, Ottawa, Canada, 2006.

[17] Tanjil Ahmed, A. K. M. Saifun Nabi, and Md. Sohrab Hossain, "A New Parallel Approach for Dynamic Clustering of Gene Sequences," unpublished.