International Journal of Computational Intelligence in Control

Parallel Implementation of Maximum Parsimony Using Multithreading Approach

Shashidhara H S
Research Scholar, Jawaharlal Nehru Technological University Hyderabad and
Ramaiah Institute of Technology, Bangalore
hs.shashidhara@gmail.com

Srinivasa K G
National Institute of Technical Teachers Training & Research, Chandigarh
kgsrinivasa@nitttrchd.ac.in

Siddesh G M
Ramaiah Institute of Technology, Bangalore
siddeshgm@msrit.edu

Abstract

Analysis of molecular sequences is the cornerstone of computational biology. It is not uncommon in bioinformatics to provide more efficient computational tools for sequence analysis. Two of the computations which are most favorite among bioinformaticians are sequence alignment and phylogenetic problems. Classic sequence alignment and phylogenetic tree construction problems quickly run out of control both in terms of memory and time of execution when either the length of individual sequences or number of sequences are increased, sometimes even an additional sequence is sufficient to break computation barrier, especially in case of phylogenetic tree construction. Here in this paper there is an attempt to both decrease the execution time and reduce the memory usage in phylogenetic tree reconstruction. In phylogenetic tree reconstruction, distance based UPGMA is combined with parsimony method to find the most accurate among competing topologies. Many techniques are used to reduce both time of construction and number of topologies like removing non informative sites which do not provide any additional information during tree construction but add to time complexity. Finally, a successful attempt is made to create the parallel version of this UPGMA and parsimony combination.

Keys - computational biology, phylogenetics, UPGMA (Unweighted Pair Group Method with Arithmetic Mean), maximum parsimony, distance based phylogeny.

1. Introduction

In addition to analysis of molecular sequences to find the functionality using sequence alignment, relationship between sequences can also be established to find their sequence of evolution and common ancestors. This study is then represented using tree structures. There are two types tree representations in such studies – taxonomy and phylogeny. In taxonomy, the classification of species is done using external features like physical characteristics, place of living, appearance similarities etc. In phylogeny, molecular sequences either nucleotide sequences or amino acid sequences, are analyzed and evolution is determined. Inferring evolutionary history using molecular sequence analysis is more reliable than just looking external features and classifying organisms. In phylogeny, the relationship is represented using phylogenetic tree[1].

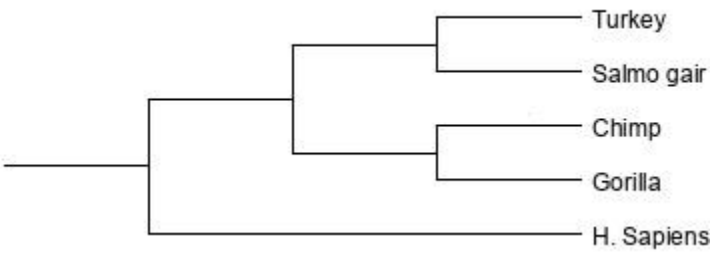


Fig1. Rooted phylogenetic tree

Each interior node represents the common ancestor. In the fig.1 above, representing a phylogenetic tree depicting the relationship between five species, Turkey and Salmo gair have a common ancestor. It is not possible to guess the relationship between these two organisms using taxonomy method. Each tip of the tree denotes the actual species participated in this phylogenetic tree construction. As evolution make changes at gene level and organisms pass this information from generation to generation, new species are originated when enough changes accumulate. This is the point where classifying the organisms based on physical characteristics start to become tedious. The evolution of organisms become apparent only through the study of their genes.

There are mainly two methods of creating phylogenetic sequences – distance based methods [2] and parsimony [3]. In distance based methods like UPGMA (Unweighted pair group methods with arithmetic mean) [4], tree is constructed by finding the number of character replaced between sequences. This number is used to assign branch lengths so that larger the length more is the time taken for evolution. In parsimony method, among the possible topologies, the one contributing minimum number of changes in the overall construction phylogenetic tree is considered the one representing the actual relationship between species whose sequences are involved in analysis.

In both the methods of constructing phylogenetic trees, a term called distance is used in the work proposed. It simply denoted the number of character changes from one sequence to another. For the following set of sequences, the simple distances can be calculated by comparing one sequence with another and tabulating the results:

```
5 42
Turkey        AAGCTTGGGC ATTTCAGGGT GAGCCCGGGC AATACAGGGT AT
Salmo gair    AAGCCTTGGC AGTGCAGGGT GAGCCGTGGC CGGGCACGGT AT
H. Sapiens    ACCGGTTGGC CGTTCAGGGT ACAGGTTGGC CGTTCAGGGT AA
Chimp         AAACCCTTGC CGTTACGCTT AAACCGAGGC CGGGACACTC AT
Gorilla       AAACCCTTGC CGGTACGCTT AAACCATTGC CGGTACGCTT AA
```

Fig 2. Format of Sequences used while constructing phylogenetic tree

The format used for reading the sequences is – the first line indicates the number of sequences participating in the phylogenetic tree construction followed by number of characters in each sequence. From next line onwards, the sequences are represented with first word indicating species name followed by actual nucleotide or amino acid sequences. If the sequences contain very large number of characters, the same format is followed keeping once species below another without continuing extended characters into next line. The species have to be present one below the other without break. The extended characters are placed below after all species are represented in the same order and the procedure continues for further characters.

Table1. Simple Distance Matrix

	Turkey	Salmo gair	H. Sapiens	Chimp	Gorilla
Turkey	0	12	19	25	26
Salmo gair	12	0	16	18	21
H. Sapiens	19	16	0	24	21
Chimp	25	18	24	0	8
Gorilla	26	21	21	8	0

Branch lengths are assigned to every branch in the topology. A branch length indicates the number of state changes from a node of the tree to another node. a state change is simply the number of character changes between two species or between species and its ancestor as shown by Table 1. But this may not be the actually represent the number of state changes because of reverse mutation happened somewhere in between the species under comparison. State A might have been changed to C in a descendent and would have changed back to A in the next descendent.

Many models are proposed to find exact substitution numbers [5]. T. Jukes and C. Cantor realized that if there are many changes in between two aligned sequences, then simple number representing the state changes between the sequences is not sufficient and a good means to measure actual number of substitutions. They proposed a model based on their hypothesis each nucleotide has an equal opportunity to be replaced by one of the other three nucleotides.
Their model suggested the following equation to find the actual distance between two sequences which according to them is more accurate compared to number of state changes:

$$d_{xy} = -\frac{3}{4}\, N\, ln\left(1 - \frac{4}{3}\frac{d_h}{N}\right)$$

Where N is the length of each sequence x and y and d_h is the number of simple state changes.

The following table (Table 2) represents the Jukes-Cantor distances which is calculated using the above formula:

Table2. Corrected Distance Matrix

	Turkey	Salmo gair	H. Sapiens	Chimp	Gorilla
Turkey	0	15.11	29.11	49.71	54.98
Salmo gair	15.11	0	22.34	26.69	34.61
H. Sapiens	29.11	22.34	0	45.21	34.61
Chimp	49.71	26.69	45.21	0	9.23
Gorilla	54.98	34.61	34.61	9.23	0

The proposed methods use this branch length table.

With the available distance matrix, the tree can be built such that the sum of lengths on each branch is equal to the distance between the sequence. A tree topology which gives the branch lengths in such a way that the sum of squared distances of sequences involved is minimal is considered the most accurate tree. This is a tedious task as it involves considering all possible topologies. There are heuristic searches available which do not guarantee the trees with smallest squared errors but are fast in pointing out the near to correct topologies.One of the most popular heuristic algorithms is UPGMA.

Algorithm1: UPGMA

Let D be the distance matrix
Do {
    Find the two sequences/groups with the smallest distance (A, B)
    Join the sequences and form a group (AB)
    Make the group parent of the two sequences joined
    Update D with new distances from group to other sequences $d_{(AB)C} = ½ (d_{AB} + d_{BC})$
} while (no further group can be done)

UPGMA assumes branch lengths on both sides of the internal node is equal which means the species involved have constant evolution time from parent which is not the case most of the times. But the distance matrix created by UPGMA is used in creating the initial tree in Parsimony. UPGMA given branch lengths for the leaf nodes. The tree in parsimony is constructed by combining the most distant sequence because in parsimony the tree with least number of changes is the best topology.

2. Related Work

UPGMA is a distance based algorithm for the construction of phylogenetic trees. It produces ultrametric trees where the leaf nodes are equidistance from the root. Y. Chen et.al. [6] have proposed parallelization of basic UPGMA algorithm using graphics processor NVIDIA Tesla C2050. Compared to its sequence counterpart, they have obtained about 95 times better speedup. The implementation uses parallel tree reduction algorithm to obtain minimum distance every iteration. UPGMA algorithm assumes equal distance from parent to offspring evolved out of the parent. But this is not the case most often. Lineages often take varying times for their evolution. In our proposal, UPGMA is used only to set initial bounds so that construction of tree can be quickly pruned.

Andrew has tried several techniques such as generating initial scores, IPC etc. [7] while trying to parallelize generating most parsimonious tree. But in the end none of them are successful in providing performance gain. The reasons listed are lack of availability of modelling data in the memory and lack of proper technique in modelling the basic parsimony algorithm. The work proposed in this paper clearly models the data and uses branch and bound modelling of the basic parsimony algorithm.

Some hardware based methods are proposed for high performance phylogenetic analysis. Server Kasap et. al. [8] used FPGA design and implemented the nodes on FPGA based supercomputer Maxell. The hardware is an array of 20 processing elements each of which is acting on a different tree topology. It can support 12 taxa but it claims can be scaled easily. It uses Sankoff's algorithm for

the construction of topology but in our proposal we use Fitch Algorithm for the same and phylogenetic reconstruction is done on a PC and no special hardware is proposed.

An XMP based parsimony search is proposed by W. Timoty et. al. [9]. It makes use of work stealing algorithm so that the work can be scaled in a distributed environment to hundreds of CPUs. A set of worker processes are created which request work from the master process. The master process actually starts creating the tree and send some portion to the worker thread. As it is indicated by them this may lead to imbalance in the solution time by processes. Also, as the work gets subdivided, it may reach a stage where the sub problems assigned to processes may become so small that communication between master and worker threads itself might become overhead.

ExactMP [10] is an efficient Parallel Phylogenetic tree reconstruction tool using maximum parsimony. The upper bound to begin constructing the topology is a four stage approach in this proposal. Rather a stringent method is followed to make sure that upper bound is very tight. It uses EDG greedy algorithm [11], followed by IEDG, TBR algorithms [12]. In our proposal we use a heuristic approach to decide upper bound and start constructing the tree and improvise based on obtained branch lengths.

The load balancing problems faced in ExactMp is addressed in a work proposed by R. Luling et. al. [13]. It demonstrates the efficiency on a network up to 256 transputers. The branch and bound is fully distributed such that each processor uses same algorithm but trying to solve different part of the problem. A significant improvement in speedup is obtained compared to efficient sequential counterpart.

Purdom et. al. [14] have proposed an optimization technique for quickly finding the most parsimonious tree which uses single column discrepancy heuristic and sequence addition using dynamic Max-mini order method. In traditional branch and bound, a cost is assigned which is equal to the discrepancy of partial tree. By using single column discrepancy heuristic, we can increase this cost discrepancy which is needed to attach the yet to be added sequences to the partial phylogenetic-tree by predicting a minimum additional Using Max-mini order, the search for the optimal tree can be terminated that guarantee sub-optimal solutions.

PhyloMOEA [15] proposed by Cancino W. et. al. combines maximum parsimony with maximum likelihood approaches to phylogenetic reconstructions. It is based on ParadisEO framework. The work proposes multi-objective approach as the authors feel having too many independent criteria to improve performance leads to conflicting phylogenies. They have also used heuristic approaches for reconstructing the tree.

Path-Relinking (PR) is a metaheuristic method used for complex problems [16]. It was defined by Fred Glover [17] and it is closely related to Tabu Search where Local searches applied to a problem check its immediate neighbors in the hope of finding an improved solution. Given two solutions called source and guiding, PR consists in transforming the source solution into the guiding solution by applying a series of modifications. An exploration phase is performed on a copy of the source solution under modification. The aim of PR is find the better solution by generating a path from the source to the guiding solution.

Yan has proposed an algorithm GRAPPA [18], a gene order based phylogeny reconstruction. They have also used branch and bound to prune the search space with some techniques such as rearranging sites to compute the tree lengths faster. The optimization algorithm proposed takes constant time to find the length of the new trees which are compatible with some partial trees. Efficient mutual lock mechanism is used to access the shared data. It is implemented for symmetric multiprocessors.

3. Proposed System

First the trees are constructed using parsimony method. Instead of examining each possibility, there are some proposed modification that can be used to avoid comprehensive search of the optimal topologies. Further reduction in search space is done using branch and bound method. Finally, this branch and bound method is parallelized to increase speed of obtaining the most parsimonious trees.
Parsimony technique describes the process of finding the evolutionary path that has least number of changes or mutations. It is based on the premise that mutations are rare events in the course of evolution and less the number of such events, the more accurate is the prediction. On the contrary, a topology which proposes to invoke more number of changes in less likely to depict the correct evolution path.
The tree is constructed by adding one species at a time. The process begins with a sequence and another sequence is added to producea sibling. The process begins to get options from the time when the third sequence is added. There will be a set of three topologies. The correct one is selected based on parsimony principle that the one which invokes less number of state changes is the best topology. When the fourth sequence is added there will be fifteen different possible topologies and with five sequences, there will be 105 possible topologies.
In general, there $(2N-3)! /2^{N-2}(N-2)!$ possible topologies for N species under consideration. For just 10 species, the number of topologies to be examined will be 3.45×10^7. As some more species are added, it is going to be a lot of computation to check all possible topologies generated.

The topologies are generated based on states of each site. If the number of sites is reduced, the number of trees generated will also be reduced and saves the computation time. Consider the following set of sequences:

Turkey	AAGCTTGGGC
Salmo gair	AAGCCTTGGC
H. Sapiens	ACCGGTTGGC
Chimp	AAACCCTTGC
Gorilla	AAACCCTTGC

Consider sites1, 9 and 10. All the sequences have same characters and no mutations. Hence, these sites can be omitted from computation of parsimony. Also, if we consider site number 2, somewhere in the tree, state changes from A to C. So the score is 1 irrespective of topology. The same case is at site 4 and 7. In the cases of site 5, the number of state change in all topologies will be 2. So, these sites are uninformative in the sense that they will not indicate most parsimonious tree. We will consider only those sites where there are two different states and each state appears at least twice. Such sites are referred as informative sites and are the only ones to be considered. This method of isolating informative sites from uninformative site and examining only informative sites reduces the computation time significantly. In the above example set of sites 3, 6 and 8 are informative. A total reduction of 70% is achieved in number of sites to be examined.

The sites to be considers for analysis are:

Turkey	GTTG
Salmo gair	GCTG
H. Sapiens	CGTG
Chimp	ACCT
Gorilla	ACCT

Further analysis indicates that the number of vertices to be searched to arrive at most parsimony can be reduced.

If we consider the first site of the first sequence. It is G state. If we add second to fifth sequences, then the number of state changes will be at least 2. If we add the state changes from other sites, which is 6 in the example taken, it indicates that minimum number of mutations when we consider all sequences is 6.

Turkey	GTTG
	2 2 1 1 = 6
Salmo gair	GCTG
H. Sapiens	CGTG
Chimp	ACCT
Gorilla	ACCT

When the topology is searched, which is done using depth first method, at any instance of time if the number of state changes goes beyond this minimum state changes calculated earlier, we can stop considering that topology for further analysis as it is not going to help in parsimony. Parsimony is relied on minimum number of changes.

This types of pruning the search is termed as branch and bound search and is helpful in restricting the number of trees to be considered significantly. Thus contributing signification increase in speedup. The earlier the pruning begins, the smaller is the time taken to run the program. This minimum threshold has to be found at the earliest and used to prune. This is where UPGMA algorithm is helpful. It is used to find this minimum number of state changes as it combines sequence with minimum number of changes in every step and build the tree. UPGMA can also be used in making the prune process start earlier in the construction of trees. UPGMA also gives branch length for the sequences as they are added. If the branch lengths are sorted in descending order and sequence with larger branch length is added, the pruning can begin based on whether score crosses the threshold. This process further reduces the computation time.

Algorithm2: Parsimony with branch and bound (siteSequences)

```
Let D be the distance matrix
Do {
Find the two sequences/groups with the smallest distance (A, B)
Join the sequences and form a group (AB)
Make the group parent of the two sequences joined
Update D with new distances from group to other sequences d(AB)C = ½ (dAB + dBC)
} while (no further group can be done)

N = number of sequences
listSequences = sequenceList(siteSequences)
for-each (site in siteSequences)
        delete site, if no state change
        delete site, if state change is constant
}

tree[] = generateTree(correctedDistance)
prunedTree[] = prune(tree[])

generateTree(correctedDistance) {
        if computeScore(treeInstance) >min_state_changes
```

```
                delete(tree)
        else
updateScore(treeInstance)
                return tree
}

prune(tree[]) {
        dMatrix = distanceMatrix(siteSequences)
        sortBranchlengths();
        generateTree(dMatrix)
}
```

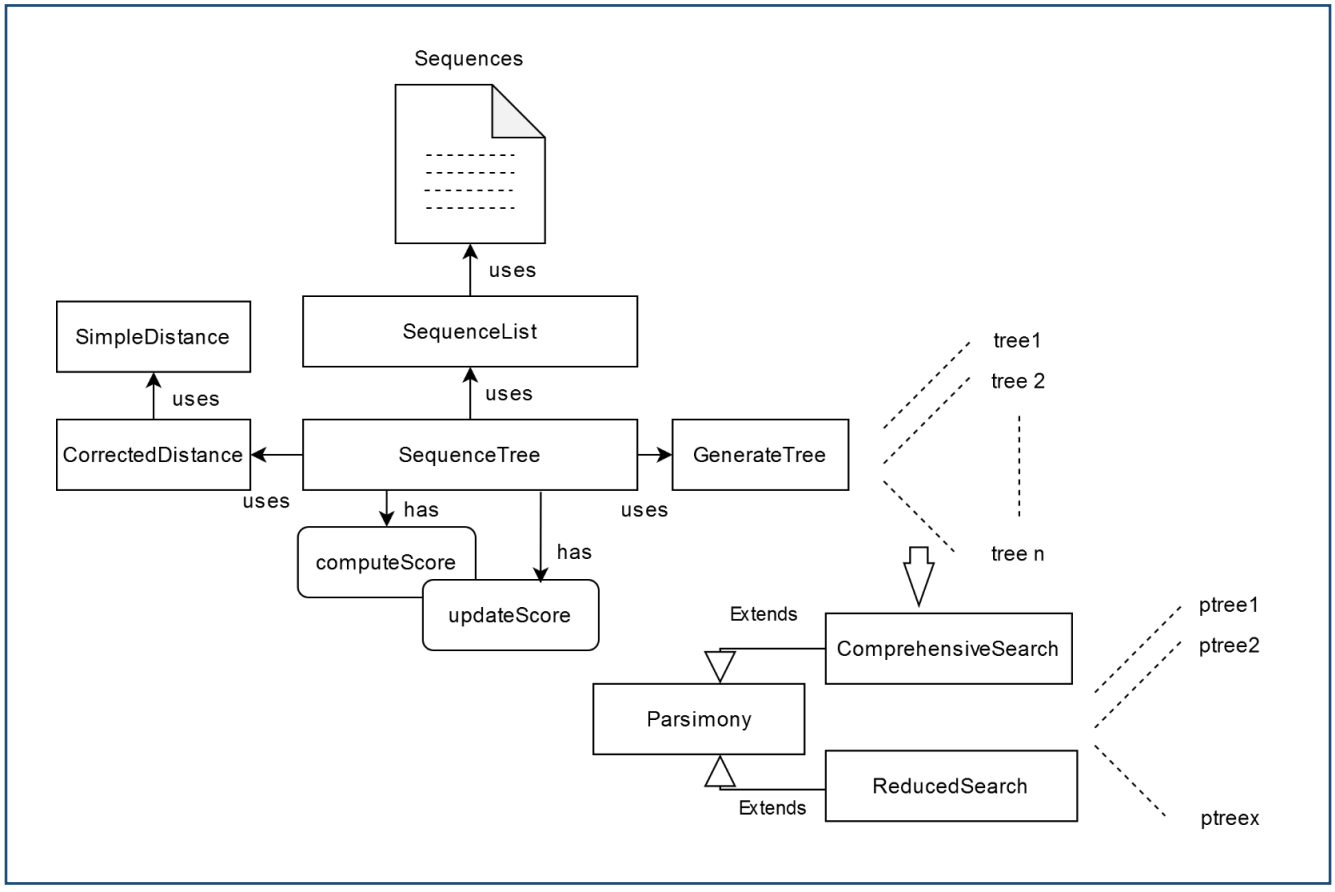


Fig 3. Maximum Parsimony Phylogenetic Tree construction

Figure 3 shows components in proposed implementation. A brief description of each component is described below:

Sequences – file containing sequences with N sites
SequenceList – A list of sequences obtained from Sequences file
SimpleDistance – has methods to find number of state changes between sequences
CorrectedDistance – uses SimpleDistance to find corrected distances because number of state changes may not be the correct measure due to reverse mutations
SequenceTree – generates trees by taking into account all pruning techniques discussed such as removal of sites with zero state changes, uninformative sites.
GenerateTree – generates trees in visual format
Parsimony – has generic methods which can be used by both ComprehensiveSearch and ReducedSearch objects.
ComprehensiveSearch – uses all sites to find the most parsimonious set of trees
ReducedSearch – used pruning technique to stop constructing topology when the total branch length exceeds minimum distance suggested by UPGMA distance matrix.

Parallel implementation of Parsimony

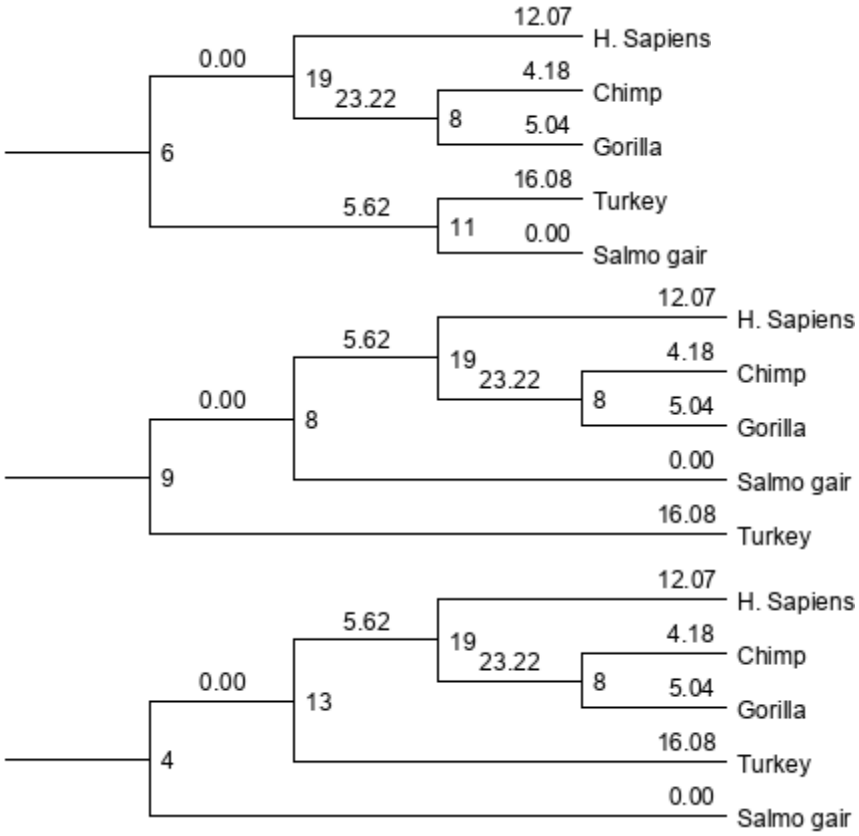
If there are N sequences involved in construction of phylogenetic tree, there are (2N-3)! $/2^{N-2}$(N-2)! possible topologies to be examined in finding the most parsimonious trees. The construction of tree begins with a character from a site in a sequence. Then it grows by combining with site from another sequence. Each level is marked staring from 1. Till level 5, there will be 105 topologies to be examined. As creating threads has overheads, it is recommended to divide the work starting from level 6 where 945 topologies need to be pruned. If the speedup obtained is not satisfactory, the division of labor can start form level 7. There are no strict rules as to when the parallelization has to start. It is arbitrary and depends upon how much efficiency is expected after parallelization. Even though the number of topologies to be assigned to a thread is divided based on number of possible topologies at a level divided by number of thread planned, each thread need not get topologies from all sites. The sites containing no state change and uninformative are all removed. So, each thread begins with a topology which is currently at the lower bound of its assigned set of topologies to be verified. There may be unbalanced work assigned as branch and bound algorithm prunes trees at different levels. It can be balanced which is not taken care in this proposal. Each thread has its own share of topologies to be verified so that no synchronization among threads is necessary at this level. But when the threads update TreeList, synchronization is necessary so that TreeList is intact. Also, there is smallest parsimony score shared variable. When a thread identifies a topology which has score less than smallest parsimony score, it has to be communicated to all other threads so that they update their smallest parsimony score to prune accordingly.

Algorithm3: Parallel parsimony
N: level at which work is divided
n: number of threads
x: minimum parsimony score

TreeList: Parsimonious tree set
for thread 1 to n
assign topology indices such that each thread gets N/n topologies to be pruned
call branch and bound instance on the site
if shared variable x has new value, adjust pruning
update TreeList

4. Experiments and Results

Following are the results and trees (Figure 4) obtained when branch and bound sequential algorithm is applied on the set of sequences mentioned in introduction sections:
Number of trees: 7
Total number of state changes: 44
Each of them have squared error of 176



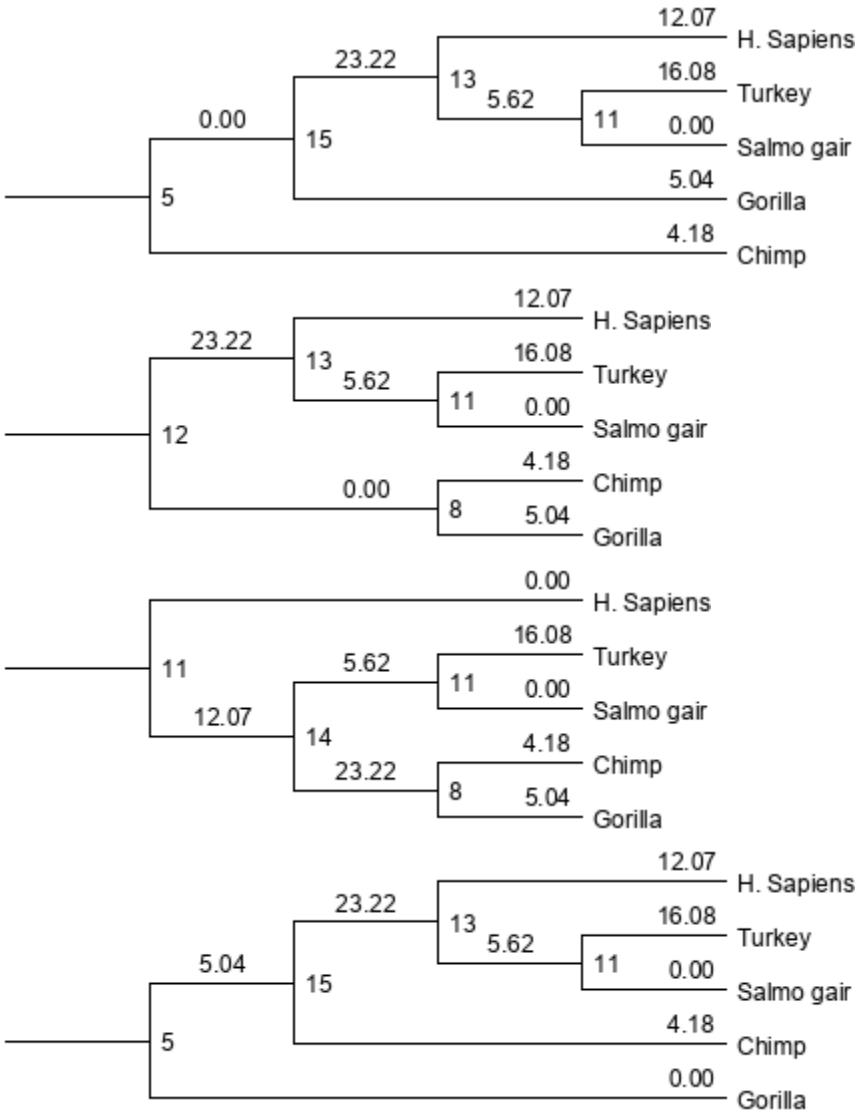


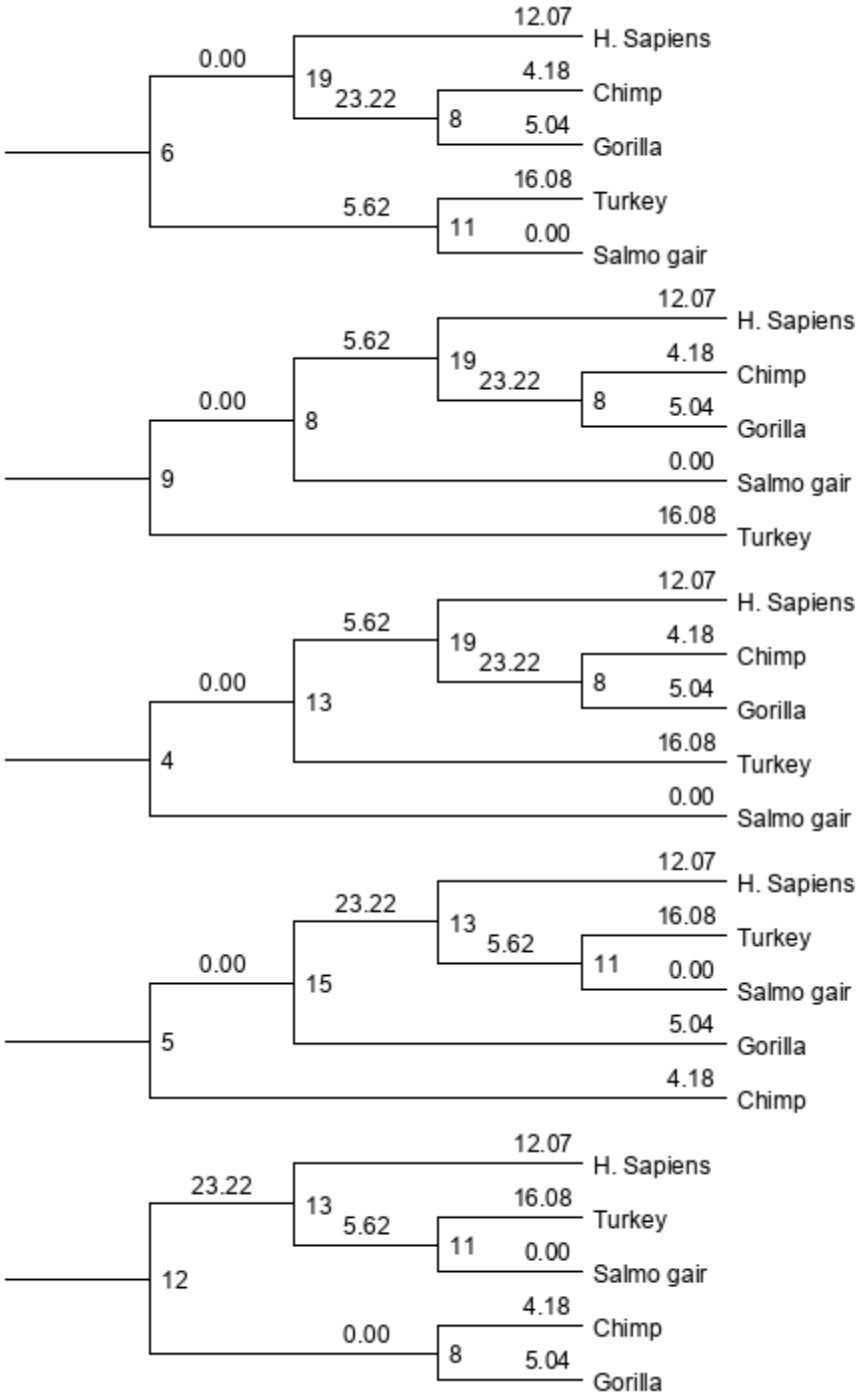
Fig. 4 The Parsimony trees generated by branch and bound sequential algorithm

Amount of time taken for preprocessing like corrected distance matrix calculation, amount of time taken for calculations like creating branches and amount of time taken for post processing like updating treelist are indicated below:

1124 msec preprocessing
53 msec calculation
490 msec postprocessing
1667 msec total

Following are the results obtained when branch and bound parallel algorithm is applied on the set of sequences mentioned in introduction sections:

Number of trees: 7
Total number of state changes: 44
Each of them have squared error of 176



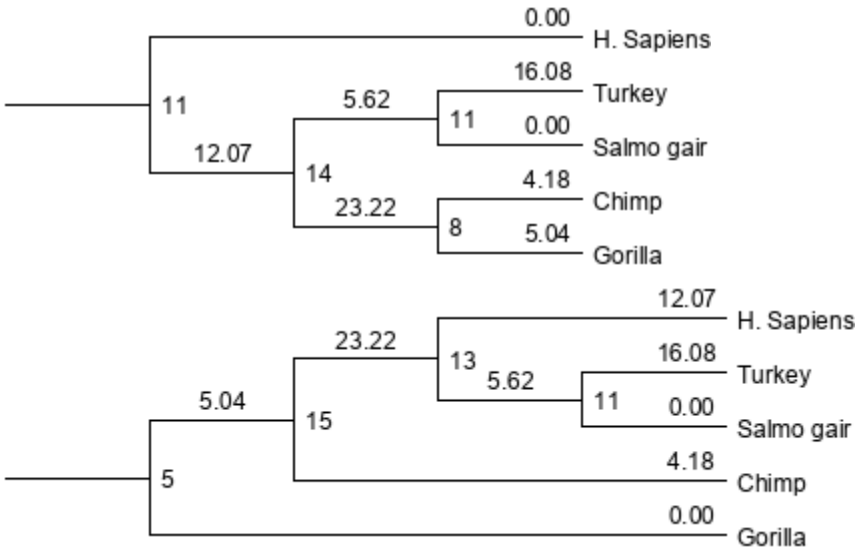


Fig.5 The Parsimony trees generated by branch and bound parallel algorithm.

The execution summary is listed below:

160 msec preprocessing
130 msec calculation
84 msec postprocessing
374 msec total

A total of 4.44 times speedup is obtained when 8 threads are used.

Table 3. Number of sequences(N) vs. Number of threads (K) vs. Execution time (T)

N	K threads	T (mS)	Spdup	N	K threads	T	Spdup	N	K threads	T(mS)	Spdup
15	seq	40495		16	seq	110750		17	seq	1439692	
	1	29526	1.37		1	121288	0.91		1	1387772	1.04
	2	15882	2.55		2	61110	1.81		2	695121	2.07
	3	11201	3.62		3	37100	2.99		3	466671	3.09
	4	8112	4.99		4	29100	3.81		4	352212	4.09
	5	7100	5.70		5	24101	4.60		5	285621	5.04
	6	6121	6.62		6	20315	5.45		6	240671	5.98
	7	5501	7.36		7	17919	6.18		7	211100	6.82
	8	5320	7.61		8	16220	6.83		8	188344	7.64

The sequential and parallel implementations for phylogenetic tree construction are applied to three sets of sequences. First set consists of 15 sequences, second et consists of 16 sequences and third set has 17 sets. The time required for both sequential and parallel implementations are listed in Table 3. The parallel implementations are applied with one to eight threads and time of execution is also listed

in Table 3along with speedup obtained. Except in the case of 16 sequences and in one case where only one thread is used, speedup is always achieved.

Figure 6 shows the speedup achieved when both number of sequences and threads are increased. As he number of thread is increased the speedup is also achieved significantly. Figure 7 also depicts how the time required for finding the maximum parsimony is reduced as the work is assigned to larger number of threads.

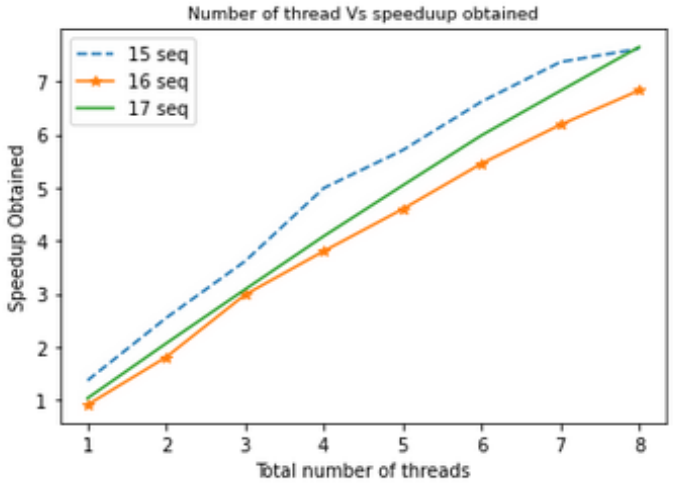


Fig 6. Speedup achieved using multiple threads

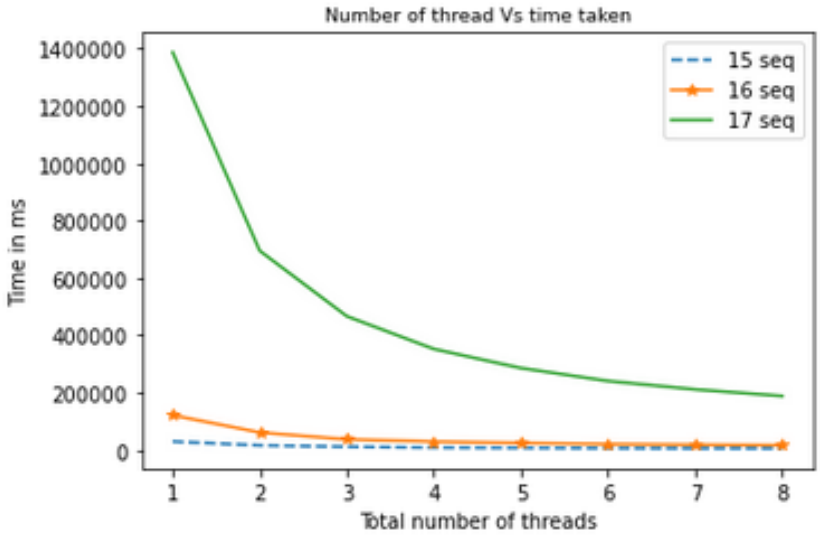


Fig 7. Actual reduction of time achieved using multithreading

5. Conclusion

Phylogenetic tree construction using parsimony technique is computationally intensive task because of number of possible topologies to be examined to arrive at the best topology. As the number of sequences increases beyond a dozen, it becomes impossible to store and verify huge amount of topologies generated. There are methods proposed to reduces number of sites to verify such as removing uninformative sites. But have little effect on reducing the computational time as the number of sequences are increased. Branch and bound algorithm is also used to reduce number of sites but helpless when number of sequences are large. Here, there is a successful attempt to parallelize branch and bound in combination with distance matrix generated by UPGMA. Distances are used to fix the maximum allowable state changes and when that threshold is crossed the topology is pruned out from further construction. The results indicate that parallelization with these modifications has significantly reduced the computation time and the results obtained from both sequential and parallel implementations are same.One modification to improve the speedup is to allocate the balanced load to every thread as against static load proposed in this work.

References

[1] Fitch, Walter M., and Emanuel Margoliash. “Construction of Phylogenetic Trees.” Science, vol. 155, no. 3760, American Association for the Advancement of Science, 1967, pp. 279–84

[2] Fabio Pardi, Olivier Gascuel. Distance-based methods in phylogenetics. Richard M. Kliman. Encyclopedia of Evolutionary Biology, Elsevier, pp.458-465, 2016

[3] Swofford, David L. "Phylogenetic analysis using parsimony." (1998): d64.

[4] Gronau, Ilan, and Shlomo Moran. "Optimal implementations of UPGMA and other common clustering algorithms." Information Processing Letters 104.6 (2007): 205-210.

[5] Erickson, Keith. "The jukes-cantor model of molecular evolution." Primus 20.5 (2010): 438-445.

[6] Y. Chen, C. L. Hung, Y. Lin, C. Lin, T. Lee and K. Lee, "Parallel UPGMA Algorithm on Graphics Processing Units Using CUDA," 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, 2012, pp. 849-854.

[7] Darling, Andrew, "Parallel implementation of maximum parsimony search algorithm on multicore CPUs" (2011). Thesis. Rochester Institute of Technology

[8] S. Kasap and K. Benkrid, "High Performance Phylogenetic Analysis with Maximum Parsimony on Reconfigurable Hardware," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 19, no. 5, pp. 796-808, May 2011.

[9] W. Timothy J. White, Barbara R. Holland, Faster exact maximum parsimony search with XMP, *Bioinformatics*, Volume 27, Issue 10, 15 May 2011, Pages 1359–1367.

[10] D. A. Bader, V. P. Chandu and M. Yan, "ExactMP: An Efficient Parallel Exact Solver for Phylogenetic Tree Reconstruction Using Maximum Parsimony," 2006 International Conference on Parallel Processing (ICPP'06), 2006, pp. 65-73, doi: 10.1109/ICPP.2006.40.

[11] R. Eck and M. Dayhoff. Atlas of Protein Sequence and Structure. National Biomedical Research Foundation, Silver Spring, MD, 1966.

[12] D. Swofford and D. Begle. PAUP: Phylogenetic analysis using parsimony. Sinauer Associates, Sunderland, MA, 1993.

[13] R. Luling and B. Monien, "Load balancing for distributed branch & bound algorithms," Proceedings Sixth International Parallel Processing Symposium, 1992, pp. 543-548.

[14] Purdom, Paul & Bradford, Phillip & Tamura, Koichiro & Kumar, Sudhir. (2000). Single column discrepancy and dynamic max-mini optimizations for quickly finding the most parsimonious evolutionary trees. Bioinformatics (Oxford, England).

[15] Cancino W., Jourdan L., Talbi EG., Delbem A.C.B. (2010) Parallel Multi-Objective Approaches for Inferring Phylogenies. In: Pizzuti C., Ritchie M.D., Giacobini M. (eds) Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics. EvoBIO 2010. Lecture Notes in Computer Science, vol 6023. Springer, Berlin, Heidelberg

[16] K. E. Vázquez-Ortiz, J. -M. Richer, D. Lesaint and E. Rodríguez-Tello, "A bottom-up implementation of Path-Relinking for Phylogenetic reconstruction applied to Maximum Parsimony," 2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM), 2014, pp. 157-163.

[17] F. Glover, M. Laguna, and R. Mart, "Fundamentals of scatter search and path relinking," Control and Cybernetics, vol. 39, pp. 653-684, 2000.

[18] Yan, Mi. High-performance algorithms for phylogeny reconstruction with maximum parsimony. The University of New Mexico, 2004.