International Journal of Database Theory and Application
Vol.10, No.1 (2017), pp.67-78
http://dx.doi.org/10.14257/ijdta.2017.10.1.07

Assessing and Mining of Phylogenetic Trees

Geetika Munjal*, Madasu Hanmandlu, Sangeet Srivastva and Deepti Gaur

The NorthCap University Gurgaon, India
munjal.geetika@gmail.com

Abstract

Assessing and Mining phylogenetic trees is very useful in storing, querying the phylogenetic databases, and finding an accurate phylogenetic tree for a set of species is very difficult. Assessing a phylogenetic tree also resolves the problem of conflicting phylogenies. This paper discusses the methods for validating and mining phylogenetic trees. We propose a new way to compare two trees by accessing importance of node in tree. This new method is applied on phylogenetic trees and the results compared with symmetric distance, Maximum Agreement Subtree and Bootstrapped tree.

Keywords*: Bootstrap, Phylogenetic, Tree mining, Sequence alignment*

1. Introduction

The phylogenetic tree represents the relationship between a set of species, and presents a model of molecular evolution Trees which is non-linear and semi-structured data structure useful and is very common in data mining problems. Any type of data, represented in the form of a tree is easy to analyze and understand. Tree mining is an component of data mining where we can extract useful information from a set of trees. A tree can be called as semi-structured data structure that shows a hierarchical relationship among its nodes. Phylogeny comprises homology and homoplacy; the former is similarity due to common ancestor while the latter is a parallel evolution [13]. The fundamental component of phylogenetics is Taxa (specie) also called as leaves and analysis of phylogenetics involves finding relationship among Taxa's. The concept of tree mining arises naturally in biology as a consistency among the evolutionary trees over the species. It is difficult to obtain a true phylogeny if we are not provided with a particular set species and a bootsrapping is promising way of gaining confidence in particular tree, and tree mining helps us gain several lines of evidence in favor of a tree.

1.1. Information on Trees [1-2]

The trees can be rooted or unrooted and the rooted tree branches emerge from one root so as to form a tree-like network. The evolutionary change and history can be visualized in a rooted tree where root depicts the ancestor, the branches lead to the descendants, and along that path, evolutionary change takes place, the species at leaf level are current day species. The trees can be ordered if the order is defined among siblings and unordered if no order is defined among siblings. In Figure 1(left) {4,5,6,7} are leaves whereas {2} and{3} are ancestors of {4} {5,6,7} respectively and {1} is a root node. In an evolutionary tree taxa is represented by its leaves and internal nodes are responsible for representing the ancestral relationships.

ISSN: 2005-4270 IJDTA

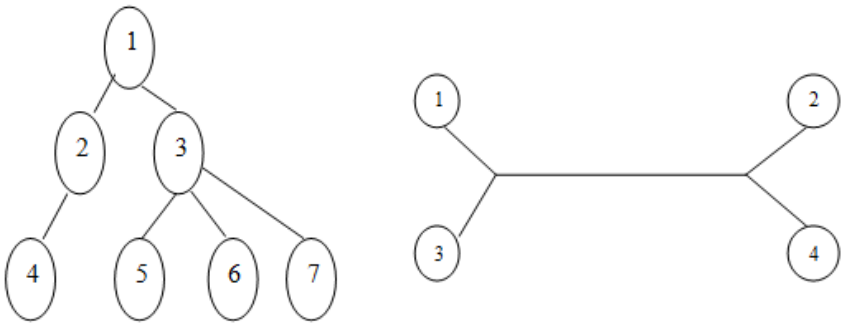

Figure 1. Rooted (Left) and Unrooted Tree (Right)

An unrooted phylogenetic tree where every internal (*i.e.* non-leaf) vertex has a degree of 3, there is no root and the edges can be spread to all sides. In the unrooted tree(right) 1,2,3,4 are leaves shown in Figure 1 which has both rooted and unrooted trees and {13}{24} are leaf pairs. Unrooted trees are also called as free trees as they make no inference about the ancestry of leaves but visualize relatedness among sequences/TAXA.

Embedded subtrees : Suppose we have a rooted tree T=(V,E,L) then T' is said to be an embedded tree of T where T' is represented by T'=(V',E',L'). Maintenance of ancestral relationship and Left to Right order of siblings are two necessary conditions of an embedded relationship. Phylogenetic tree mining is a kind of embedded subtree mining.

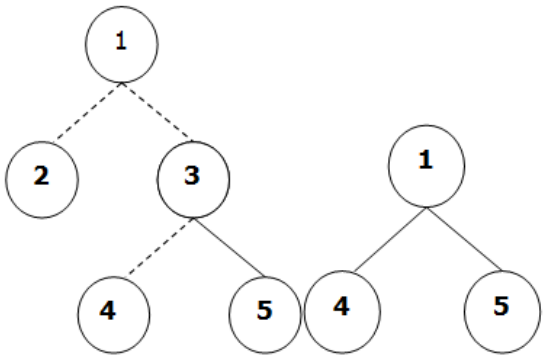

Figure 2. Original Tree (Left) Embedded Subtree (Right)

1.2. Induced Subtree [2-3]

In a tree T let the vertex set be V and edge set be E, then T' with V' and T' with E' be the vertex set and edge set respectively. These are said to be the induced subtree of T only when the labeling of E and V is preserved in T'. So an induced subtree follows Canonical ordering of siblings and removes the leaf nodes repeatedly

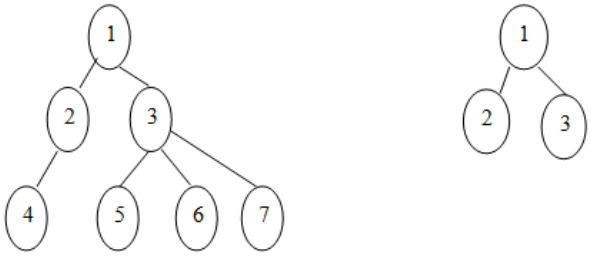

Figure 3. Original Tree (Left) Induced Subtree (Right)

Mining of the frequent patterns in a tree is aimed at identifying the common patterns [1]. A frequent subtree is a tree that is supported by at least some given number of the input trees. Frequent subtrees can reveal common patterns. The frequent tree mining aims at discovering all frequent subtrees from a large database of trees, which is referred to as forest. Support is a major component in mining of the frequent patterns and is defined as the total number of common subtree trees in the tree database. If there is one appearance of a particular subtree in a forest then the support count is 1 and if appearance increases its support count is also increased. A minimum threshold for the count is defined by the user and a subtree whose support is greater than the user defined threshold then that tree is said to be frequent. There are various frequent pattern mining algorithms. All the algorithms follow the strategy used in the well known Apriori Algorithm that is based on the iterative pattern mining where we break each iteration into two phases: i) *Candidate Generation:* Frequent patterns discovered in one iteration are used to generate potentially frequently candidates. We can merge two patterns with size k and consist of k-1 elements to generate candidates with size k+1. ii). *Support Counting:* In this phase we find the support of the frequent candidates, ignore the less frequent candidates and keep the actually frequent candidates. Actually frequent candidates are those whose support is greater than the user defined threshold. The scope list based method for frequent subtree was introduced by Zaki [1] which follows the representation of the occurrence list and an efficient candidate generation method. The goal of finding the frequent subtrees in phylogenetics is to find a subset of the species on which all or some significant species tree matches.

2. Phylogenetic Tree Building Methods [14-15]

Various tree building methods are characterized as either distance based methods or character based methods. Distance based methods include unweighted pair grouping using arithmetic mean(UPGMA) and neighbor joining and character based methods include maximum parsimony and maximum likelihood. Distance based methods are based on sequence similarity and are computationally fast but there are chances of information loss due to data transformation. However they are good for large datasets [14]. Character based methods require data in the form of the aligned sequences and are widely used for inferring phylogenies. They include maximum parsimony method which aims at finding the maximum parsimonious tree from all possible trees. This method determines the mutations between the species to find the length of trees, and selects a tree having the shortest length. The outcome of the maximum parsimony method is an unrooted tree with the maximum parsimony score. Parsimony method has two motives: i) To construct all possible trees, and. ii) To find a tree having the largest length by an algorithm. Problems of maximum parsimony method is that the tree space is very huge as it generates unrooted tree, *i.e*. one tree with 3 possible leaves and 3 trees with 4 possible leaves. As the number of species increases, the possible unrooted trees increase very rapidly. So it is not possible to do an exhaustive search on large datasets. This problem is solved by branch and bound method which randomly constructs a tree by adding a specie and cutting the lineages

which are not compatible (depending on the length of tree) with the tree. It follows a shortcut for perfection. In the Maximum likelihood method all possible branch lengths and topologies are examined to obtain the likelihood of a tree. Finding the best suitable tree for a small set of nodes is not very costly but as the set of nodes increases the numbers of possible trees increase exponentially and the task becomes intensive, so the next important thing is to find an optimal tree.

3. Validating Phylogenetic Trees

3.1. Bootstrapping Phylogenetic

Bootstrap is a sampling technique where the sequence data is re-sampled to create a new alignment matrix of identical dimension, *i.e*. each bootstrap replicate contains the same number of species. From each replication (artificial data), a tree can be reconstructed using any of the available reconstruction techniques discussed above. Bootstrap support values can be determined for every branch that connects two inner nodes of the tree, that is, for each inner branch of the tree [5]. Support is a measure to give a score to each clade among all bootstrap replicates. As many bootstrap trees are generated through repeated sampling and the bootstrap score (or support) of a branch in the inferred tree is computed as the proportion of the bootstrap trees that contains this branch/clade. As finding a support value is our primary task in bootstrap analysis, so some fast method is required to compare support, and also to explore the effect of using increasing numbers of replicates on support values. Moreover bootstrap reflects the results based on an alignment matrix and constructed tree if a there is some noise in the matrix and tree construction method then it will lead to the incorrect assumptions [5]. The bootstrap results are dominated by the alignment results and a wrong alignment will lead to misguiding the bootstrap value. An alternate to bootstrap is Jack-knife methods but it works on smaller datasets as it doesn't duplicate the original dataset, but considers the randomly selected samples from the dataset. Both bootstrapping and Jacknife suffer from the problem of time complexity for large datasets. As the number of replicates increases fast bootstrap can be used but the bootstrap value becomes imprecise [5]. In spite of problems associated with bootstrapping it is a preferred method to find the stabilized groups in a tree.

3.1 MAST

MAST step involves, for each of the clusters finding a majority accepted subtree (MAST) as a consensus tree representing the overlap of the cluster. MAST is a strong representative of the common Substructures as each MAST is embedded as a subtree in each tree in the input collection; it does not give information about relationships among species [6]. MAST is a representative of common substructures and it follows the concept of embedded and isomorphic trees The MAST problem is polynomially-solvable for two trees, but is NP-hard for three or more input trees. The complexity of MAST is O(n log n) and its variant has a complexity of $O(K\ n^3 + n^3)$ where k and n are the number of binary rooted trees and the number of taxa respectively. MAST is also known as common pruned tree and any of the pruned trees will be identical to the strict consensus tree of the pruned input trees. MAST returns a set of trees and can also be used to summarize the results of bootstrap [7]. All these trees need to be stored somewhere to choose the optimized one by applying backtracking procedure. This entire process may lead to a combinatorial explosion problem.

$$MAST(T1, T2) = \mathrm{T1}(\text{if } \mathrm{T1} = \mathrm{T2}) \text{ else } \mathrm{MAST(T1,x)U\ MAST(T2,y)}$$

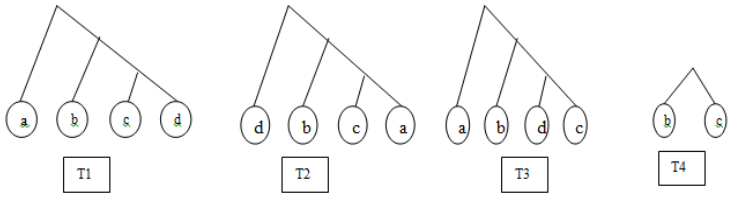

Figure 4. Maximum Agreement Subtree is T4 of T1,T2 and T3

4. Proposed Word

4.1. Distance Algorithm for Comparing Two Trees

We propose an optimized solution for comparing two trees where we can prune each specie to reach for the best results after analyzing the pruning effect. To know the pruning effect on a tree the bootstrap tree needs to be parsed again. This comparison metric tells whether the two trees fall under the category of similar trees or dissimilar trees and then by pruning the edges one at a time we can find which edge is responsible for the maximum distance.

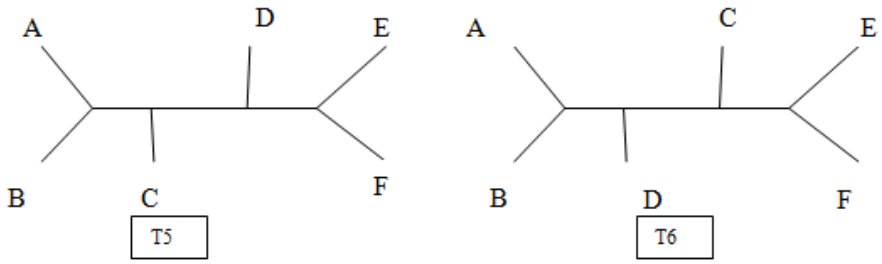

Figure 5. Figure To Illustrate Distance Algorithm for Comparing T5 and T6

Distance between each possible combination of nodes is calculated and after that the edges are pruned

Table1. Distance among Tree T_5 and T_6

Nodes	Distance (T1)	Distance (T2)	T1 –T2
AB	2	2	0
AC	3	4	1
AD	4	5	1
AE	5	3	2
AF	5	5	0
BC	3	4	1
BD	4	3	1
BE	5	5	0
BF	5	5	0
CD	3	3	0
CE	4	3	1
CF	4	3	1
DE	3	4	1
DF	3	4	1
EF	2	2	0

Table 1 gives the node-wise distance between the trees T_5 and T_6 . Counting the number of nodes having the same distance, where the difference is either 0 or 1 the number of nodes having different distance can be found. If the nodes having the same distance is greater than the latter then the two trees can be put in a similar category otherwise they are considered dissimilar. As can be seen in Table 1 the total number of possible combination of nodes is 15, number of nodes having the same distance is 6 and number of nodes having different distances is 9. Here the number of nodes having different distances is greater than the nodes having the same distance and distance is 9/6, *i.e.* 1.5. So these two trees can be considered dissimilar. But applying tree pruning, *i.e.* eliminating some portion of the tree we can get very useful results. Sometimes, the tree becomes so large that a specific part of the tree is not useful at all. Eliminating that part does not affect the overall results of the tree and it is better to prune that part. As shown in Figure 6 pruning the connecting edges As of the trees T_5 and T_6 one at a time we obtain the trees T_7 and T_8 respectively and then apply the distance algorithm.

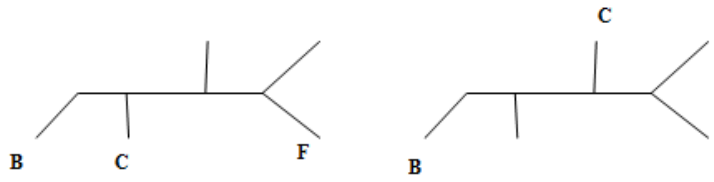

Figure 6. Resultant Trees after Pruning Edge A from T5 and T6

We have removed the edge A from both the trees; now applying the distance algorithm on the above trees we obtain the distances between the nodes (BC) as 3 and 4 respectively for the trees T3 and T4. We computes the distances among all nodes (BD, BE, BF, CD, CE, CF, DE, DF, EF) in T3 and T4. Here the possible combination of nodes is 10 and the number of nodes having the same distance is 6. The number of nodes having a different distance is 4. So from the above results, we can say that pruning edge A has no positive effect. The total nodes are 10 and the distance between the tree after pruning edge A is 6/5, *i.e* 1.2. The results of pruning each edge from T1 and T2 are shown in Table 1. By pruning edges C and D, there is a positive effect because the nodes having a different distance are greater.

Table 2. Reflecting the Effect of Pruning in Trees in T5 and T6

Pruned edge	Number of Nodes having same distance	Number of Nodes having different distance	Distance	Pruning effect t on tree
A	6	4	1.2	neutral
B	4	6	1.2	positive
C	6	4	0.8	neutral
D	4	6	0.6	positive
E	6	4	1.2	neutral
F	5	5	1.2	neutral

Table 2 shows the distances between trees T3 and T4 after pruning each edge. So the average distance comes out to be 1.06

Algorithm finaldist

Input: Tree $t_1(u,k)$ & $t_2(v,k)$// u and v are the edges and k is number of node
Output: finaldist //distance between two trees
dist array $|t_1,t_2|$ //array of distance between two trees t_1 and t_2
finaldist=0,dist[u,v] = 0
for i=1 to k *//For each vertex v in tree t_1 and t_2*
by pruning i each edge in t_1 and t_2
for j=1 to n-I // where n is the number of edges
$dist_j[u,x] = \sum w_1(u,x) + w_2$ // where w_1 is weight assigned in tree $\mathbf{t_1}$ and w_2 assigned in tree $\mathbf{t_2}$ between edge connecting u and x by default w is 1
end
finaldist=avg($dist_I[u,x]$)
Stop

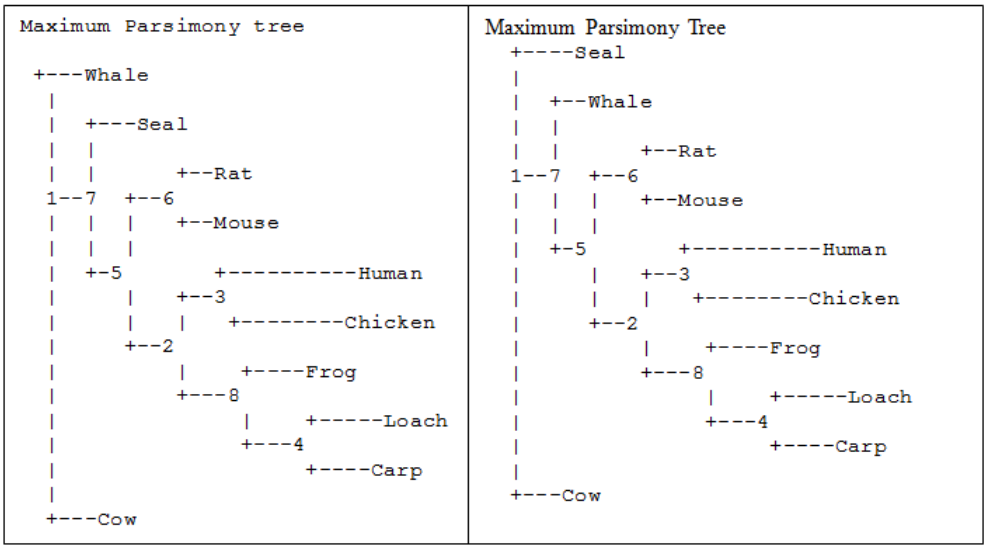

Figure 7. Maximum Parsimony Tree Generated Using Phylip T_7 (Left) T_8(Right)

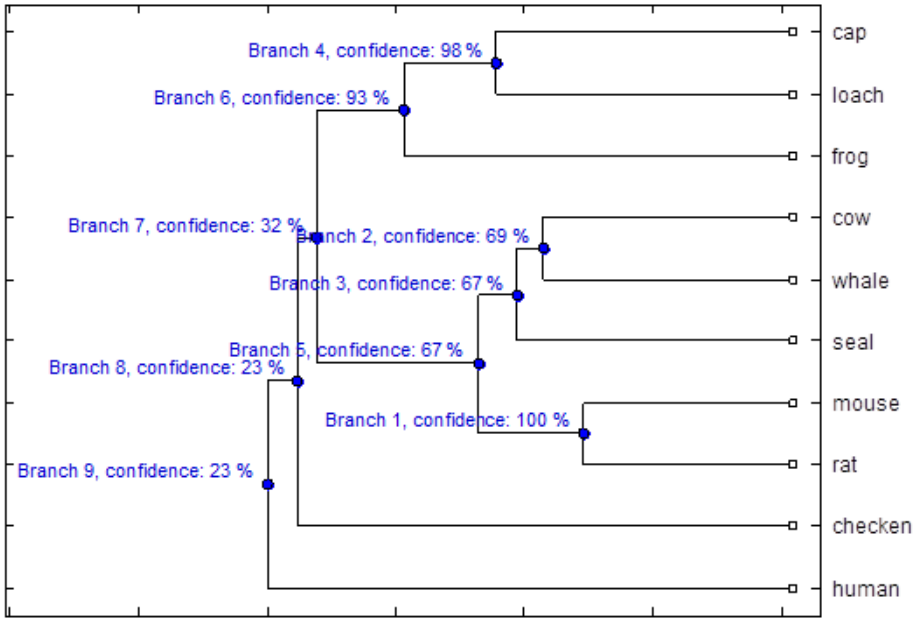

Figure 8. T9 Bootstrapped Tree with Confidence Values

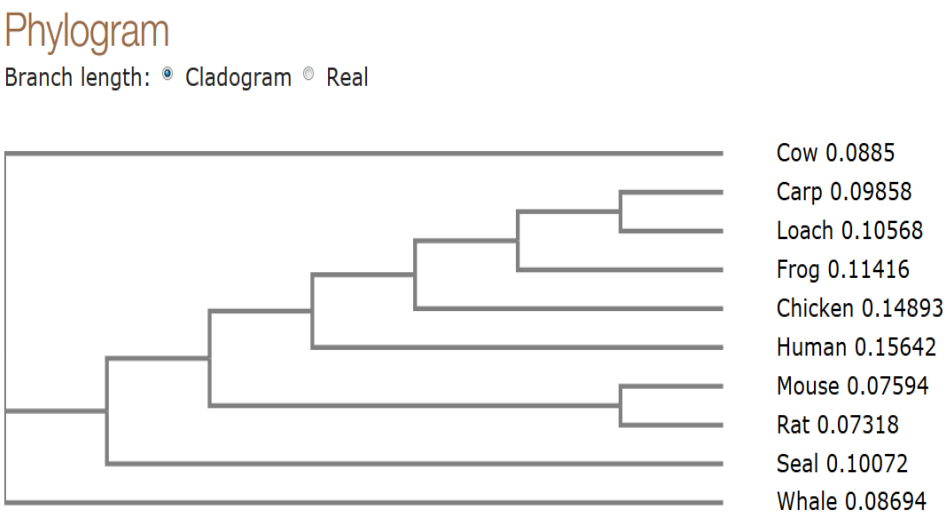

Figure 9. T10 guide Tree Generated by ClustalW

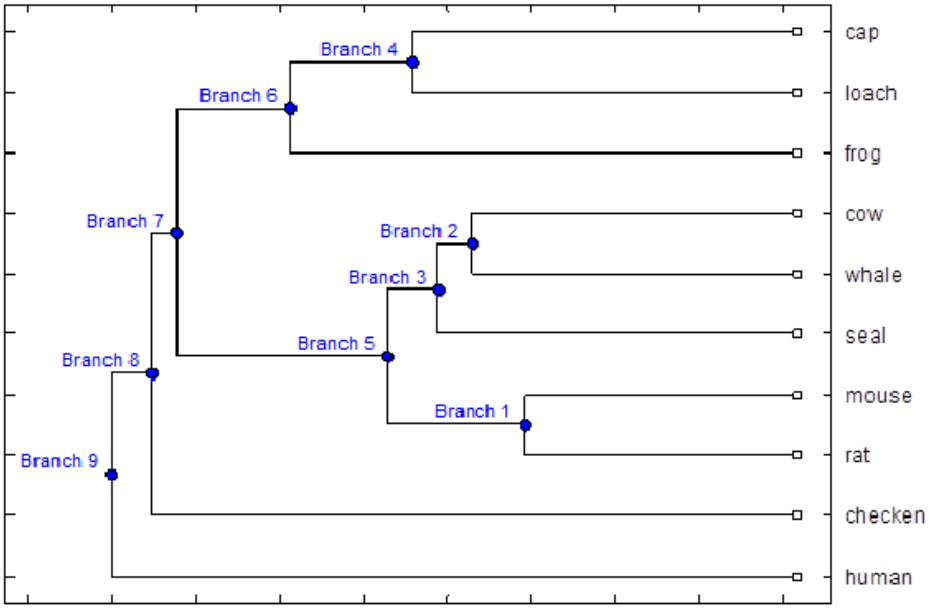

Figure 10. T11 Jukes Cantor Distance

Table 3. Depicting Various Distances between Trees with Respect to Jukes Cantor Tree(T_9)

Distance	T_{12}(maximum likelihood)	T_7 (maximum parsimony)	T_8 (maximum parsimony)
Branch Score	.41	.494	.492
Symmetric Distance	4	2	0
Distance algorithms	.2	.2	.2

Table 4. Depicting Distances between Trees with Respect to Guide Tree ClustalW

Distance	T_{11}	T_7	T_8
Branch Score	.161	.132	.130
Symmetric Distance	2	4	2
Distance algorithms	.4	.5	.5

Tables 3 and 4 give the difference among the trees. The symmetric distance is calculated using treedist [11] module of PHYLIP which is based on Robinson foulds [16] metrics. The branch score distance [17] uses branch lengths, and can only be calculated when the trees have lengths on all branches. The distance of clustal W [12] tree (Figure 9) and Jukes–Cantor tree (Figure 10) is 0.3 and If we remove any edges connecting the nodes { Mouse, rat, loach, Carb, frog} from Jukes-cantor tree t_9 its effect will be neutral but removing the edges connecting the nodes {human, chicken}{seal, cow, whale} will give positive results and also decrease the distances among the trees. As can be seen in the bootstrap tree branch connecting {loach, Carb, frog} and { Mouse, rat} have confidence value of 98% and 93%, whereas branch connecting {human, chicken} {seal, cow, whale} have confidence value 23%. Here the similarity between the trees is depicted as the distance between by assigning weight of 1 to each edge. Table 3 shows the maximum similarity with the clustalW tree using branch score and distance algorithm also it has the same set of nodes directly connected to each other, thus ignoring the topological dissimilarity.

5. Conclusions

This paper investigates the problem of tree comparison and validation. The similarity indicates how the distance algorithm has been effective in finding the significant branches in a tree. Note that the distance between the two trees gives a measure of similarity. For instance higher the distance more dissimilar the trees are.

Although Bootstrap has been the most widely used method for phylogenetic tree validation but as the dataset increases the confidence values may not signify the tree accuracy. MAST can be very useful for comparing two trees but being topology based it fails on one to one comparison. Moreover none of the methods gives information regarding the significance of branch, for which we have introduced pruning of each node one by one. The distance algorithm can be very informative if two trees share the same set of species but with different topologies. We have applied distance algorithm specifically for phylogenetic trees but the method can be a useful tool in general tree structured database as well.

References

[1] Zaki, Mohammed Javeed. "Efficiently mining frequent trees in a forest: Algorithms and applications." IEEE transactions on knowledge and data engineering 17.8 (2005): 1021-1035.
[2] Aida Jimenez, Fernando Berzal, and Juan Carlos Cubero “Mining Diff erent Kinds of Trees: A Tree Mining Overview” CEDI, September 2007, pp 343-352.
[3] Tatsuya Asai1, Hiroki Arimura1, Takeaki Uno, and Shin-ichi Nakano,”Discovering Frequent Substructures in Large Unordered Trees”, Conference on discovery science, 2003, pp. 47-61.
[4] John Bluis and Dong-Guk Shin, “Nodal Distance Algorithm: Calculating a Phylogenetic Tree Comparison Metric” Computer Science and Engineering University of Connecticut Storrs, CT 06269-3155, USA ,Bioinformatics and Bioengineering,IEEE,2003,PP 87-94.
[5] Yu Lin, Vaibhav Rajan, Bernard M.E.Moret “A metric For Phylogenetic Trees Based On Matching” IEEE, Volume 9, July 2012, pp 1014-1022.
[6] Hong Huang and Yongji Li “MASTtreedist: Visualization of Tree Space based on Maximum Agreement Subtree” Journal of Computational Biology, Jan 7,2013, PP 42-49.

[7] Vincent Berry & Franc ois Nicolas “Maximum agreement and compatible supertrees” Journal of Discrete Algorithms, Volume5, September2007,pp 564-591
[8] Hasan H. and Khalid S. “A new sequence distance measure for Phylogenetic tree construction” OxfordJournal for Bioinformatics vol 19, pp 2122-2130, 2003.
[9] Damian Bogdanowicz and Krzysztof Giaro “Matching Split Distance for Unrooted Binary Phylogenetic Trees” IEEE/ACM Trans. on Computational Biology and Bioinformatics, Volume 9 Issue 1, January 2012 pp 150-160.
[10] Bryant, David, et al. "Computing the quartet distance between evolutionary trees." Symposium on Discrete Algorithms: Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms. Vol. 9. No. 11. 2000.
[11] Felsenstein, Joseph. "{PHYLIP}: phylogenetic inference package, version 3.5 c." (1993).
[12] “ clustal omega” http://www.ebi.ac.uk/services
[13] Bernard M. E. Moret and Tandy Warnow, “Reconstructing Optimal Phylogenetic Trees :A challenge in Experimental Algorithmics” LNCS springer 2547, pp. 163–180, 2002.
[14] Jeffrey Rizzo and Eric C. Rouchka, “Review of Phylogenetic Tree Construction” University of Louisville, Bioinformatics review(report) 2007.
[15] R.Potter, Constructing Phylogenetic Trees using Multiple Sequence Alignment, Master thesis, University of Washington 2008
[16] Robinson, DavidF, and Leslie R. Foulds. "Comparison of phylogenetic trees." Mathematical biosciences 53.1 (1981) pp.131-147.
[17] Kuhner, Mary K., and Joseph Felsenstein. "A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates." Molecular Biology and Evolution 11.3 (1994) pp. 459-468.
[18] Munjal, Geetika, Madasu Hanmandlu, and Deepti Gaur. "A New Alignment Free Method for Phylogenetic Tree Construction." International Journal of Database Theory and Application 8.6 (2015): pp. 111-124.
[19] Geetika, Hanmandlu, M., and Deepti Gaur. "Analyzing DNA Strings using Information Theory Concepts." Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies. ACM, 2016.
[20] Hanmandlu, M., Ashish Sani, and Deepti Gaur. "Modified k-Tuple Method for the Construction of Phylogenetic Trees." Trends in Bioinformatics 8.3 (2015): 75.

