A HEURISTIC ALGORITHM FOR OPTIMAL AGREEMENT SUPERTREES

M. A. HAI ZAHID
Dept. of E&EC
IIT-Roorkee 247667
zaheddec@iitr.ernet.in

ANKUSH MITTAL
Dept. of E&CE
IIT-Roorkee, 247667
ankumfec@iitr.ernet.in

R.C. JOSHI
Dept. of E&CE
IIT-Roorkee 247667
joshifcc@iitr.ernet.in

ABSTRACT
Phylogenetic supertree is a collection of different phylogenetic trees combined into a single tree forming tree of life. Many smaller overlapping phylogenetic trees are combined in such a way that no branching information is lost. There may exist an exponentially large number of supertrees for a given set of trees. The optimal tree is a selected based upon different optimality criteria. In this paper we present a polynomial time heuristic algorithms for merging the trees, checking their optimality based on the ordinary least square criteria. The out come of the algorithm is an optimal agreement super tree.

Keywords:
Phylogenetic tree, phylogenetic supertree, heuristic algorithm, optimality criteria, ordinary least square

1. INTRODUCTION
Phylogeny, a tree of life of all the lineages on the earth, provides a framework to facilitate biological information retrieval and prediction. Presently most of the individual researchers or teams are concentrating on the evolutionary pathways of specific phylogenetic groups.

Moreover it does not seem to be possible for an individual researcher or a small team to construct a phylogenetic tree of life consists of 1.7 million described species. Phylogenetic supertrees will provide a modest solution to this problem. Phylogenetic supertree is a result of combining many smaller overlapping phylogenetic trees in such a way that no branching information is lost. Supertree methods are useful because most of the research teams concentrate on the phylogenetic study of a specific class of species and due to certain bias some taxa will be studied rigorously. All the smaller phylogenetic trees, studied by different groups, can be combined together using supertrees concept to construct the "TREE OF LIFE", which classifies all the species.

Computer science plays an important role in finding the optimal supertree. Many heuristics can be suggested for the construction of supertrees because it is computationally expensive to construct an optimal agreement supertree with exhaustive search technique.

The problem can be defined as follows: let $T=\{T_1, T_2, \ldots, T_k\}$ be a set of unrooted trees, where each T_i is a distinctly leaf labeled with leaf set $L(T_i)$, the leaf set may overlap. An agreement super tree of T is an unrooted tree T with the leaf $L(T_1) \cup L(T_2) \cup \ldots \cup L(T_k)$ such that each tree T_i is an induced subtree of T.

Several methods have been proposed for the combining binary, rooted phylogenetic trees. A plenary survey of phylogenetic supertree construction methods is given in [1][2]. Henzinger et al. [3] showed how to

modify the total agreement supertree problem for any T in such a way that it can be solved in $\min\{O(Nn^{0.5}), O(N + n^2 \log n)\}$ time where $N = \sum_{T_i \in \mathsf{T}} |T_i|$; a similar kind of work is done by Jesper et al. [4]. A polynomial time algorithm for constructing supertrees from phylogenetic distances was proposed by Stephen in [5]. On the other hand the total agreement problem for unrooted trees is NP-Hard [6]. A polynomial-time algorithm for computing an unrooted total agreement supertree if one exist when all K input trees are binary and $k = O(1)$ was given by Bryant in [7], which gives the best complexity $O(4^k n^{2k+1})$ when binary character compatibility score is considered as optimality criterion with respect to a given weightings of splits. In this paper we present a heuristic algorithm for the construction of supertree from unrooted, phylogenetic trees. The paper is organized in sections, where in section 2 we presented the basic definition required to understand the problem and mathematical theorems used for supertree construction. Section 3 is dedicated to algorithms developed based upon the mathematics defined in section 2. In section 4 we discuss the experiments carried out.

2. METHODOLOGY

2.1. Basic Definition Related To Supertrees

In this section we give some of the basic definitions which are necessary to understand the agreement supertree problem.

An unrooted phylogenetic tree is a finite, acyclic connected graph with all the internal nodes have the degree $d \geq 3$, and the leaf nodes are labeled uniquely by the members of the leaf set of the tree $L(T)$. A tree T' is said to be induced tree having vertices $L(T')$ and whose edges are the edge set consist of those edges of T with having incidents from $L(T')$. An agreement supertree is a collection of trees $\{T_1, T_2, ..., T_k\}$ with leaf sets $L(T_1), L(T_2), ..., L(T_k)$ is an unrooted tree T with the leaf set $L(T_1) \cap L(T_2) \cap ... \cap L(T_k)$ such that each tree T_i is an induced tree of T.

A vertex of degree one in a rooted or unrooted tree is called a *pendent vertex* (i.e. leaf node) and the edge incident only on one pendent vertex is called *pendent edge*. If a pair of pendant vertices is adjacent to a non-pendent vertex, which is not adjacent to any other pendant vertices then the pair is called *pendent pair*. Pendant pairs should be preserved while merging two trees for optimality.

2.2 Split Constrained Agreement Supertree

Given a phylogenetic tree T, removing an edge from it divides the leaf set of the tree into two parts called a split of T. A single split is represented by A|B where $A, B \subseteq L(T)$ and $A \cap B = \phi$; and total splits in a tree are represented as *splits(T)*.

Splits constrained optimization can be used to construct optimal agreement supertrees [7]. Which formally can be defined as follows: let S be the set of splits on leaf set L of tree T and the trees are degree bound d. the tree T with degree bound d and $splits(T) \subseteq S$ are said to follow the split constrained optimization. This criterion can be used for the construction of phylogenetic super tree as follows: let $\mathsf{t} = \{T_1, T_2, ..., T_k\}$ be a collection of trees, and $L = L_1 \cup L_2 \cup ... \cup L_k$, where $L_i = \mathsf{L}(T_i)$. The splits of T can be defined as:

$$S(\mathsf{T}) = \{A_1 \cup A_2 \cup ... A_k \,|\, B_1 \cup B_2 ... \cup B_k\}$$

Where

$$A_i \,|\, B_i \in splits(T_i) \cup \{\phi \,|\, L_i\}, i = 1,2,...,k$$

and

$$(A_1 \cup A_2 \cup ... A_k) \cap (B_1 \cup B_2 \cup ... B_k) = \phi$$

Here the assumption is $A_i \,|\, B_i \in splits(T_i)$ implies $B_i \,|\, A_i \in splits(T_i)$. Let $n = |L_i|$, for each tree T_i there are at most $2n - 3$ splits and in set of splits of all the trees, which participate in super tree, the total number of splits are $O(2^k n^k)$, where k is number of trees and n represents total number of leaf in all the trees.

A theorem given in [7] to find whether a given tree T is a phylogenetic supertree of T or not states as follows:

Let T be an unrooted phylogenetic tree with leaf set L. if each tree $T_i \in \mathsf{T}$ is an induced subtree of T then $splits(T_i) \subseteq S(\mathsf{T})$.

The proof and other details of the above theorem are given in [7]. This theorem express, that any agreement supertree T for T satisfies that $splits(T) \subseteq S(\mathsf{T})$. The number agreement supertree can be exponentially large, to find the optimal tree we used the ordinary least square optimization criterion. It can be calculated on the basis of the distance between the taxa when tree were not merged, represented as d, and distance between the taxa after forming supertree, represented as p. The sum of least squares can be calculated as

$$\|p-d\|^2 = \sum_{x \in L(T)} \sum_{y \in L(T)} (d_{xy} - p_{xy})^2$$

We wish to find the tree with minimum OLS score.

3. ALGORITHMS AND EXAMPLES

In this section we give the algorithms for merging two trees in such a way that no branching information is lost.

We designed two cases for the merging of two trees. They are, the trees with non-overlapping sets of leaf nodes, that is $L(T_i) \cap L(T_j) = \phi$; and trees with overlapping sets of leaf nodes, that is $L(T_i) \cap L(T_j) \neq \phi$. This can be generalized to k trees.

Trees with non-overlapping sets of leaf nodes:

Let two unrooted trees T_i and T_j have non-overlapping leaf nodes, $L(T_i) \cap L(T_j) = \phi$, the optimal agreement super tree can be obtained by merging these two trees with a new edge. The procedure is to first convert T_i and T_j to rooted trees by first removing an edge and then joining two subtrees with a new node called root node (ρ). The edge incident to the adjacent node of the pedant pair is the most suitable candidate to be removed. This will split the unrooted tree into two unrooted subtrees. A new vertex x called root is added to merge two subtrees to produce rooted version of the given unrooted tree.

If more than two pendant pairs found then the edge which splits the tree into two subtrees with almost equal pendant pairs is removed; finally a root node is added to it, which gives the rooted version of the tree.

An example for the conversion of unrooted binary tree to rooted binary tree is shown in Figure 1and Figure 2. Here Figure 1 shows an unrooted binary tree that is to be converted to rooted binary by the method described above. In this example all capital letters (A, B, C…) represents vertices and small letter (a, b, c, d…) are used to represent edges. According to the heuristic formed edge e is the suitable edge to be removed show in Figure 2.

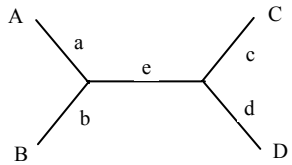

Figure 1. Unrooted binary tree T to be converted to rooted binary

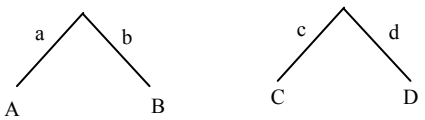

Figure 2. Unrooted tree T after removing an edge A th e.

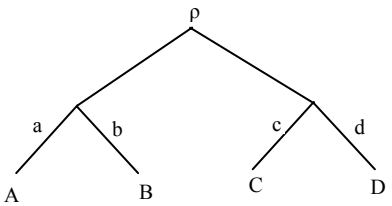

Figure 3. Rooted tree T' of unrooted tree T after adding a new node ρ.

The algorithm for the conversion of unrooted binary tree to compatible rooted binary tree will take the unrooted tree as input and

produces it compatible version of rooted tree. The compatibility between trees is discussed in [9]. The algorithm is as follows.

ALGORITHM:UNROT_TO_ROT(UnrootedT-ree T)
Begin
If number of pendants is even
Begin
For each split in SPLITS (T)
If the split divides the pendants equally then
Add a new vertex R
Add new edges from the nodes, which had the removed edge incident to them, to the new vertex.
Break;
Else
Continue;
End
Else If number of pendants on one side is
Begin
For each split in SPLITS (T)
If the split divides the pendants equally then
Add a new vertex R
Add new edges from the nodes, which had the removed edge incident to them, to the new vertex.
Break;
Else
Continue;
End// end else
End // End of Algorithm

The rooted trees are constructed from unrooted trees in such a way that all the pedant pairs are preserved.
The merge operation for the given two unrooted trees with non-overlapping leaf node sets will makes use of the above algorithm. The algorithm for merging two unrooted, non-overlapping phylogenetic trees is given below. The result of experiment is given in section 4.

ALGORITHM: MERGE_NO (Tree T_1, Tree T_2)
Begin
$T_1^{'} = UNROT\_TO\_ROT(T_1);$
$T_2^{'} = UNROT\_TO\_ROT(T_2);$
Add a new edge between the roots of $T_1^{'}$ and $T_2^{'}$
End // end of algorithm

Trees with overlapping leave nodes:
Given two trees T_i and T_j have overlapping leaf nodes sets, $L_i \cap L_j \neq \phi$, can be merged by first making a rooted tree for each non overlapping set of leaf nodes then adding the overlapping edges in a way to reduce the optimality criterion. The separation of overlapping and non-overlapping leafs is as follows.

$$L_i \cap L_j = C_{ij}$$
$$L_i - C_{ij} = NO_i$$
$$L_j - C_{ij} = NO_j$$

Where C_{ij} is common or overlapping leaf nodes in trees T_i and T_j. NO_i and NO_j represent non-overlapping leaf nodes in T_i and T_j.
The first step in merging is to merge the non-overlapping nodes using the algorithm described for merging non-overlapping trees. The second step is to add the overlapping leaf nodes on by on to reduce the ordinary least square optimization criterion.
The algorithm for the trees with overlapping leaf nodes is given below.

ALGORITHM: MERGE_OVLP (Tree T_1, Tree T_2)
Begin
$C_{12} = L_1 \cap L_2;$
$NO_1 = L_1 - C_{12};$
$NO_2 = L_2 - C_{12};$
$T_1^{'} = T_1\big|_{\{CN_1\}};$
$T_2^{'} = T_2\big|_{\{CN_2\}};$
$T_1^{''} = UNROT\_TO\_ROT(T_1^{'});$
$T_2^{''} = UNROT\_TO\_ROT(T_2^{'});$

$C_{12}^{'} = UPGMA(dist\_of\_C_{12});$
Add a new common vertex v
Add an edge between the root of $T_1^{''}$ and v

Add an edge between the root of T_2'' and v

Add an edge between the root of C_{12}' and v

End// end of algorithm

Here we used UPGMA algorithm for the construction of the tree for overlapping leaf nodes [8]. There are three distance measures can be considered they are maximum $\{d_i(l_1, l_2)\}$, where d_i is the distance between l_1 and l_2 of tree T_{ij} and $\{l_1, l_2, \ldots l_k\} \in C_{12}$; minimum $\{d_i(l_1, l_2)\}$; and mean $\{d_i(l_1, l_2)\}$. The results of the experiments are given in section 4.

1. EXPERIMENTAL EVALUATION

In this section we discuss the results of the experiments conducted upon two different simulated data sets. The first experiment consists of a set trees with non-overlapping leaf node set, and second experiment is carried out on the set of trees with overlapping leaf node set. The distance between the taxa (leaf nodes) is considered to be the minimum number of edges between them. The experiments are as follows.

EXPERIMENT 1:

In this experiment set of two trees are a given as input, $\tau = \{T_1, T_2\}$, which have non-overlapping leaf node sets. The set of leaf nodes of tree T_1 and T_2 is $L(T_1) = \{A, B, C, D\}$ and $L(T_2) = \{E, F, G, H\}$ respectively and $L(T_1) \cap L(T_2) = \phi$. The trees are represented as graphs with no cycles. Trees T_1 and T_2 are shown in Figure 4(a) and Figure 4(b).

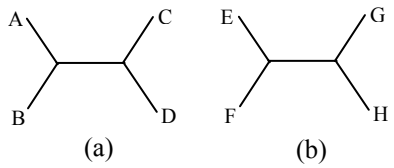

Figure 4. Trees T_1, T_2 to be merged

This set is given as input to the algorithm MERGE_NO that makes use of the algorithm UNROT_TO_ROT to covert the unrooted trees to compatible rooted tree then an edge is added between the roots of the tree. The result is as shown in Figure 5.

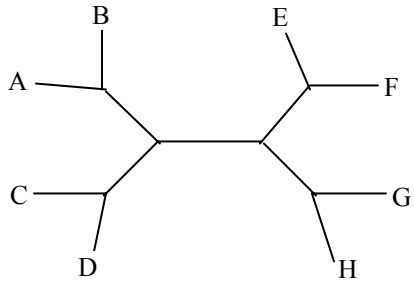

Figure 5. Resulting super tree after merging trees T_1 and T_2.

The ordinary least square criterion calculated for the above tree, where the minimum number of edges are considered as the distance between the taxa, is 80 based upon optimality criteria given above. Other optimal trees can also be obtained but this method gives the closely optimal tree with polynomial computation time.

EXPERIMENT 2:

In this experiment set of two trees are a given as input, $\tau = \{T_1, T_2\}$, which have overlapping leaf node sets. The set of leaf nodes of tree T_1 and T_2 is $L(T_1) = \{A, B, C, D\}$ and $L(T_2) = \{A, B, C, E\}$ respectively and $L(T_1) \cap L(T_2) = \{A, B, C\}$. The trees are represented as graphs with no cycles. Trees T_1 and T_2 are shown in Figure 6(a) and Figure 6(b).

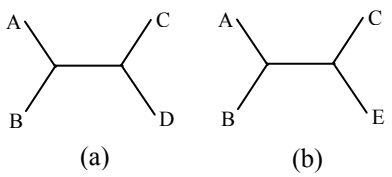

Figure 6. Trees T_1, T_2 to be merged

This set is given as input to the algorithm MERGE_OVLP that makes use of the

algorithm UNROT_TO_ROT to covert the unrooted trees to compatible rooted trees of the non-overlapping edges. A rooted tree is constructed using UPGMA for overlapping taxa. And finally all the trees are merged using a common vertex. The result is shown in Figure 7.

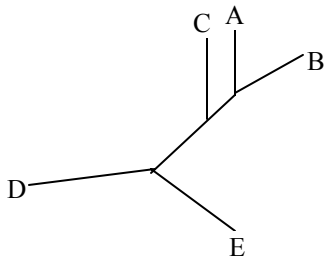

Figure 7. Resulting super tree after merging trees T_1 and T_2.

The value of ordinary least square criterion for the above example is 10.

5. CONCLUSIONS

In this paper we developed a heuristic algorithm for merging of two trees. We considered the distance between the taxa (leaf nodes) is the minimum number of edges between them. The same algorithms can be used for the trees with the weights on the edges, the results will be much more accurate than the reslts without edge weights. If the trees are constructed using different methods, which assigns different weights to the edges, leads to conflicts. This algorithm have the search space of only $\sum_{i=1}^{k}(2n_i - 3)$ states, where as the algorithm designed by Bryant [7] have $(2^k n^k)$ states, here n and k are number of leaf nodes and number of trees respectively. This reduced the search space to a great extent and thus reduces the complexity of the algorithm.

REFERENCES

[1] O. Bininda-Emonds, J. Gittleman, and M. Steel. "The (super) tree of life: Procedures, problems, and prospects", *Annual Review of Ecology and Systematics*, 33:265–289, 2002.

[2] M. J. Sanderson, A. Purvis, and C. Henze. "Phylogenetic supertrees: assembling the trees of life", *TRENDS in Ecology & Evolution*, 13(3): 105–109, 1998.

[3] M. R. Henzinger, V. King, and T. Warnow. "Constructing a tree from homeomorphic ubtrees, with applications to computational evolutionary biology", *Algorithmica*, 24(1):1–13, 1999.

[4] J. Jansson, H.-K.Ng Joseph, K. Sadakane and W. Sung, "Rooted maximum agreement supertrees", M. Farach-colton (Ed.): LATIN 2004, LNCS 2976, pp. 499-508, 2004.

[5] J.W. Stephen, "Constructing rooted super trees using distances", Department of Mathematics, Iowa State Univ. April, 2004.

[6] M. Steel, "The complexity of reconstructing trees from qualitative characters and subtrees", *Journal of Classification*, 9(1):91–116, 1992.

[7] D. Bryant, "Optimal agreement supertrees", In *Proc. of the 1st International Conference on Biology, Informatics, and Mathematics* (JOBIM 2000), volume 2066 of *LNCS*, pages 24–31. Springer, 2001.

[8] P.H.A. Sneath and R.R. Sokal. Numerical Taxonomy. W.H. Freeman, San Francisco, 1973.

[9] D. Bryant. *Building Trees, Hunting for Trees, and Comparing Trees: Theory andMethods in Phylogenetic Analysis*. PhD thesis, Univ. of Canterbury, N.Z., 1997.