www.icgst.com

Least Common Ancestor Based Method for Efficiently Constructing Rooted Supertrees

M. A. Hai Zahid, Ankush Mittal, R. C. Joshi
Department of Electronics and Computer Engineering, IIT-Roorkee
Roorkee, Uttaranchal, INDIA
[zaheddec, ankumfec, joshifcc]@iitr.ernet.in,
http://www.iitr.ernet.in

Abstract

Phylogenetic supertree is a collection of different phylogenetic trees combined into a single tree forming a tree of life. The smaller overlapping phylogenetic trees are combined in such a way that no branching information is lost. This problem is important for several biological applications. Yet the solution is difficult as exponentially large number of supertrees exists for a given set of trees and the optimal tree has to be selected based upon some optimality criteria. In this paper, we propose a polynomial time algorithm for combining phylogenetic trees, which makes use of least common ancestor information as optimality criterion. The algorithm satisfies the desirable properties of a phylogenetic supertree method as mentioned in literature, and constructs a single phylogenetic supertree even for incompatible input trees which is difficult to solve**.** Experimental results and comparisons with other works show the superiority of our algorithm.

Keywords: *phylogenetic tree, phylogenetic supertrees, optimality criteria, least common ancestor, and compatibility.*

1. Introduction

Phylogeny, a tree of life of all the lineages on the earth, provides a framework to facilitate biological information retrieval and prediction. Genome sequencing projects produce the molecular data without analysis, but the evolutionary history (phylogenetic tree) relates organism and genes. Some of the primary applications of phylogenetics include, understanding the interaction between the genes, drug design, predicting gene functionality, predicting origin and spread of diseases, and origins and migrations of humans. Currently 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 consisting of 1.7 million known species. Phylogenetic supertrees provide a modest solution to this problem.

Many efficient phylogenetic reconstruction methods, such as Maximum Parsimony and Maximum Likelihood, are available. However, these methods lead to hard optimization problems and are limited to small number of taxa. Using these methods, more accurate phylogenetic trees can be constructed for small number of taxa in a reasonable time frame. On the other hand, methods like distance-based phylogenetic reconstruction can be used for large taxa, but the conversion from sequence data to distance data leads to loss of information. Thus, a method which exploits the features of distance and character based phylogenetic reconstruction methods is required to construct phylogenetic supertree.

Phylogenetic supertree construction methods follow a divide and conquer strategy, to combine trees consisting of hundreds of taxa in them. As a first step smaller trees are constructed using efficient phylogenetic reconstruction methods, and then in the second step, these smaller trees are combined into a large tree consist of all the taxa such that preserve the phylogenetic information carried by each tree is preserved.

In general, if the taxa in the small trees are overlapping, the result of combination will be supertree. When the input trees have the same leaf set then the result of amalgamation of the small trees will be a consensus tree. A good survey of consensus methods and supertree methods is given in [8] [13] respectively. There exist many efficient consensus tree algorithms [19], but very few algorithms for supertree methods are published with the following properties [1]:

1. The supertree should be computed in polynomial time.
2. The algorithm should preserve the branch information shared by all the input trees.
3. Changing the order of the input trees should not change the resulting supertree.

4. Renaming or relabeling should lead to the change in the corresponding label in the resulting tree but should not change the structure.
5. If all the input trees are compatible the algorithm should return a tree which displays all the input trees

Currently the most widely used phylogenetic supertree method is Matrix Representation of Parsimony (MRP) [2][3]. It encodes the input trees into binary characters and the supertree is constructed from the resulting data matrix using a parsimony tree building method. It does not satisfy the first property because finding the most parsimonious tree is NP-compete [4]. Only few algorithms are published with the above desirable properties that include: Mincut [5], Modified Mincut [6], and Rankedtree [7]. All the methods, mentioned above, results in a supertree for the rooted collection of input trees. On the other hand, Steel [12] proved that the problem of combining unrooted phylogenetic trees is an NP-hard. A heuristic algorithm for combining unrooted trees is given by Zahid [9]. The super tree method for weighted input trees is given in [21].

In this paper we propose a new technique for the supertree construction for rooted trees, which exhibits all the above desirable properties. We proposed a new distance measure and a variant of standard UPGMA algorithm to compute the supertree in polynomial time. The least common ancestor algorithm given in [22] is used for the detection of the least common ancestors of different species. Other algorithm for finding least common ancestors are given in [23].

The paper is organized as follows. Section 2, covers some basic terminology required to understand the problem and desirable properties. In section 3, we present the new method for the construction of the supertrees. And in final section we discuss the experiments carried out and compare them with the Mincut [5] and Adam's consensus method [17]. Adam's consensus method is compared for trees on common leaf only.

2. Preliminaries

Rooted phylogenetic tree and rooted triplets

A tree is an acyclic connected graph and can be represented as $T=(V,E)$. A vertex $v \in V$ is internal if the degree of v is greater than two, otherwise v is leaf and a distinct vertex with degree two is called the root. An edge e, connected to the vertices u and v, is internal if both u and v are internal vertices, otherwise it is an external edge. The set of leaf nodes of a tree is represented as $L(T)$.

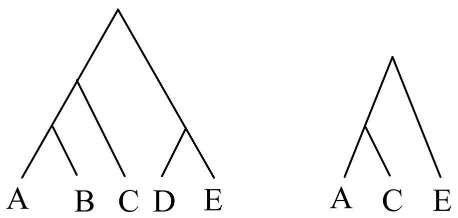
A *phylogenetic* or *evolutionary tree* is a tree T havinge a single internal node with degree two and rest of the internal nodes have degree three or more. If the degree of each internal node is three except root then it is called a binary rooted tree. Two rooted phylogenetic trees on set of species Y, $T'=(V', E')$ and $T''=(V'',E'')$ are considered as identical if there exist a bijection function $\alpha: V' \rightarrow V''$, which includes a bijection from E' to E'' and fits the set of leaf nodes Y. Except the root, the labeling of the internal nodes is unimportant in phylogenetics. The internal nodes represent the hypothetical ancestors of the descendents.

A rooted triplet is a tree with three leaf nodes. On the leaves a, b and c, a triplet can be represented as (ab)c or ab|c , if the path from the leaves a and b to root node does not intersect the path from c to root node, and a, b shares a least common ancestor.

Restriction on phylogenetic trees

Let T be the rooted tree with the leaf set X. If the set X' (subset of) X is given, the topological restriction of T to X' is the tree obtained by deleting the nodes which are not in the path from root to any node in X' and contracting the internal edges incident on vertices with degree two. An example of the topological restriction, represented as $T'=T|X'$, is shown in Fig. 1. T' is called the induced subtree of T by X'.

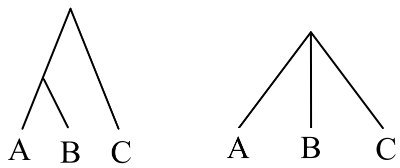
Compatibility of phylogenetic trees



(a) (b)
Fig.1. Let T be the tree in (a), then (b) represents T|{A,C,E}.

A rooted phylogenetic tree T displays a rooted phylogenetic tree T' if T' can be obtained from an induced subtree of T by contraction. It is denoted as $T' \leq T$.

According to Steel [5], a collection of rooted phylogenetic trees is compatible if there is a phylogenetic tree that displays all of them. According to Wormald's [14] two trees are compatible if and only if one is a subtree of other. Fig. 2 shows T' and T'' in which T' is not a subtree of T'' but T'' can be obtained from T' by contraction. According to Steel T' and T'' are compatible. In this paper we use the Steel's definition of compatibility.



(a) Tree T' (b) Tree T''
Fig 2: Tree T' and T'' are compatible according to Steel M.

Tree reconstruction based on distances

Given sequence data of the species, the distance between the species is calculated based on the number of mutations between a pair of species. These evolutionary distances are then used to reconstruct phylogenetic tree. There are many algorithms available, which make use of the distance data to find the phylogenetic tree such as, Unweighted Pair Group with Arithmetic mean (UPGMA) [10], and Neighbor Joining (NJ) methods [11].

In this paper we make use of a variant of the standard UPGMA algorithm in which the distance matrix is based on the average number of edges between two species. UPGMA It is basically a clustering algorithm that group two smaller clusters iteratively until the root of the final phylogenetic tree is reached.

3. UPGMA and its variant

In this section we describe the standard UPGMA algorithm and its failure with respect to the constraints on the tree distances and finally we present a variant of the standard UPGMA.

Standard UPGMA

This method is statistical based and the simplest among the phylogenetic tree construction methods. The method is as follows:

1. Take distance matrix as input
   For example:

Species	A	B	C
B	d_{AB}	-	-
C	d_{AC}	d_{BC}	-
D	d_{AD}	d_{BD}	d_{CD}

2. Species with the smallest distance will be clustered.
3. The new distance matrix is computed with the distance between new group and the remaining species as
   $d_{(AB)C}$= ½ (d_{AC}+ d_{BC}) for all the species.
4. Repeat until all the species have been grouped.

Failure of Standard UPGMA

This method applied to pair wise sequence data will return a rooted phylogenetic tree. But when applied to tree distance data lead to the clustering of wrong taxa. This will be clear with the following example. The distance matrices which reflect the failures of standard UPGMA are shown in Fig. 4 and Fig. 5.

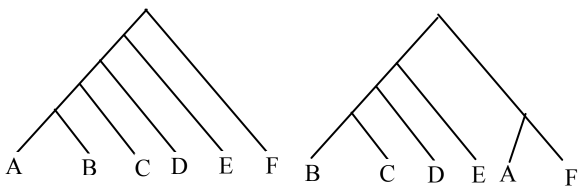


Fig. 3 (a) Tree *T'* Fig. 3 (b) Tree *T"*

The average distance matrix of the trees, shown in Fig.3 (a) and 3(b), is shown in Fig. 4.

	A	*B*	*C*	*D*	*E*
B	4				
C	4.5	2.5			
D	4.5	3.5	3		
E	4.5	4.5	4	3	
F	4	6	5.5	4.5	3.5

Fig 4: Distance matrix for the trees shown in Fig. 3.

The distance matrix in Fig. 4 shows that the distance between *B* and *C* is least and thus leads to the grouping of the *B* and *C*. the distance matrix after merging *B* and *C* is shown in Fig. 5.

	A	*BC*	*D*	*E*
BC	4.25			
D	4.5	3.25		
E	4.5	4.25	3	
F	4	5.75	4.5	3.5

Fig 5: Modified distance matrix after grouping *B* and *C*.

The species *D* and *E* at minimum distance in the matrix shown in Fig. 5, but if the distance from the least common ancestor of the *BC* is considered then the *D* is close to the group *BC* as the distance between the least common ancestor of *BC* and *D* is 2 as shown in Fig 3(a) and 3(b). Employing the standard UPGMA algorithm leads to wrong clusters, thus leading to incorrect phylogenetic tree.

UPGMA variant

To avoid the above problem the following variant of standard UPGMA is used in our algorithm.

1. Make a cluster of the species with the minimum distance between them.
2. Remove the grouped species from the input trees and add a new node with the cluster name.
3. Add an edge between the least common ancestor of the grouped species and new node.
4. Calculate the distance matrix for the new trees.
5. Repeat the first 4 step until a single group is left.

This method results in suitable tree and can be computed in $O(kl^3)$, where k is number of tree of trees and l is number of common leaf nodes in the collection of the input trees.

The new modified distance matrix, shown in Fig. 6, will be used for further grouping instead of the distance matrix show in Fig. 5.

	A	*BC*	*D*	*E*
BC	3.5			
D	4	2.5		
E	4	3.5	3	
F	3.5	5	3.5	3.5

Fig 6: Distance matrix computed using variant of standard UPGMA.

4. The supertree algorithm

In this section we describe the new technique for the construction of the supertrees and establish the desirable properties of the supertree method. The advantage of this algorithm is that it will return a tree even if the input trees carry conflicting information. The algorithm is described below.

Algorithm: DISTSUPERTREE(*T*)
Input: A set of rooted phylogenetic trees $T = \{T_1, T_2, ..., T_k\}$.
Output: A rooted phylogenetic tree on $\cup_{i=1}^{k} L(T_i)$.

1. Find common and distinct leaves in the given trees
   $COMM = \cap_{i=1}^{k} L(T_i)$, $DISTINCT_i = L(T_i - COMM)$ for *i=1* to *k*.
2. For all trees, find the restrictions of the input trees on the subsets *COMM* and *DISTINCT*.
   $T_i^{'} = T_i|\{COMM\}$ and $T_i^{''} = T_i|\{DISTINCT_i\}$ for *i =1* to *k*.
3. Construct the phylogenetic consensus tree for the for the restriction trees constructed in step 2 as $ST = UPGMA\_variant(DISTANCE(T_i^{'}))$.
   /* for each tree $T_i^{'}$ the number of edges between the leaves is considered as the evolutionary distance

between them. For the final tree $T^{'}$ the average of the distances of the trees is considered. */

4. If $DISTINCT \neq \phi$
   For each i=1 to k
   If $T_i^{''}$ is a subtree of T_i without performing contraction, then the root of $T_i^{''}$ is connected with root of the *ST*, and the whole tree is renamed as *ST*.
   Otherwise for every node N_j in *DISTINCT*$_i$*:*
   Add an edge between N_j and its least common ancestor in T_i in the tree *ST*.
5. Output *ST*.

The first step of the algorithm finds the *COMM*, common and *DISTINCT* leaf nodes in the collection of input trees. Restriction trees for the both *COMM* and *DISTINCT* sets are constructed in the second step of the algorithm. Step 3 takes the restriction trees of *COMM* set as input and uses the variant of UPGMA to find the consensus tree for the common leaf nodes. The distance between the species is considered as the average number of edges between them. Finally the distinct nodes are added to the consensus tree, which is the result of step3, based on least common ancestor optimality criteria.

Properties of the DISTSUPERTREE

The DISTSUPERTREE shows all the properties given by Steel [1] for the supertree methods. Here we establish the desirable properties for the above algorithm.

The DISTSUPERTREE method compute the supertree in polynomial time as each step in the algorithm takes polynomial time to complete the desired task. The key steps in the algorithm are 3 and 4, which computes the supertree of all the taxa. Step 3 takes $O(kl^3)$, where k is number of tree of trees and l is number of common leaf nodes, $l = |COMM|$, in the collection of the input trees. Step 4, which grafts the distinct nodes on the consensus tree resulting from step 3, will take $O(km)$, where m stands for the number of nodes in *DISTINCT* set of each input tree($m = |DISTINCT_i|$, for i=1 to k). The total complexity of the algorithm is $O(kn^3)$, where k is number of trees and n is the total number of leaf in all the input rooted trees.

It is evident that the algorithm satisfies the properties 3 and 4, which states that the result of the supertrees should not be affected if the order of the trees is changed or leaves are relabeled. The algorithm does not make use of the order or labeling information rather uses the structural topology for the construction of the supertree. Hence properties 3 and 4 are satisfied.

The fact it returns a supertree is proved in the following theorem.

Theorem 1: Let $T = \{T_1, T_2, ..., T_k\}$ be a set of rooted phylogenetic trees. Then the DISTSUPERTREE applied to *T* returns a tree.

Proof: To simplify the proof we consider two cases, the first case is of the collection of input trees classifying the common species, and the second case is of the collection of the input trees classifying the overlapping set of species.

For the case I, it is clear that the DISTSUPERTREE method returns a tree when applied to a collection of trees *T*, which classify the common leaf nodes. The algorithm returns a consensus tree at step 4. UPGMA variant is used for the construction of the consensus tree which always results in a single rooted tree.

For the second case, the input collection of the tree classify overlapping set of species, Step 4 uses the least common ancestor information carried by the input trees to ensure that all the leaf nodes are connected to the proper least common ancestor.

The leaf nodes of each input tree come under *COMMON* set or *DISTINCT*$_i$ set. Both the sets are used for the construction of the supertree. The loop in step 4 ensures that all the trees are processed for each node in the *DISTINCT* set, which is added to the resulting supertree. This leads to a single rooted supertree. □

The proof of the theorem 2 depends on the result proved in theorem 1 of [20].

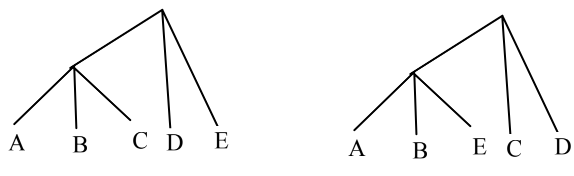
Lemma 1: let T and T' are two phylogenetic rooted trees. Then $T \leq T'$ if and only if $r(T) \subseteq r(T')$ and $L(T) \subseteq L(T')$.

Theorem 2: Let $T = \{T_1, T_2, ..., T_k\}$ be a set of compatible rooted phylogenetic trees. Then the DISTSUPERTREE applied to *T* returns a tree, which displays all the input trees.

Proof: if $T = \{T_1, T_2, ..., T_k\}$ is the set of compatible rooted collection of input trees. Let $r = \cup_{i=1}^{k} r(T_i)$ is set all the rooted triplet of the input trees, where $r(T_i)$ the set of triplets of the tree is T_i. Then by comparing the algorithm Mincut [7] with the DISTSUPERTREE, when the input collection of trees is compatible, the resulting tree of both the algorithms is identical (shown in section 5) when applied to $r(T)$. Therefore, as $r(T)$ is a subset of rooted triplets of the tree returned by Mincut, $r(T) \subseteq r(ST)$ also holds. According to lemma 1, *ST* displays all the trees in the collection *T*. □

5. Experimental Evaluation and comparisons

In this section we discuss the results of the experiments conducted upon two different trees and compare them with the Mincut, and Adams consensus methods. The first data set consists of the trees, which shares common leaf nodes. The result will be a consensus tree. The input trees for this method are given in Fig. 7 (a) and Fig. 7 (b).



(a) Tree *T'* (b) Tree *T''*

Fig. 7: Trees *T'* and *T''* are amalgamated to give a consensus tree using different methods.

If the trees in Fig. 7 are given as input to DISTSUPERTREE, after the first step, *COMM* contains all the leaf nodes and set *DISTINCT* is empty. The distance matrix computed for the trees is given in Fig. 8.

The matrix represented in Fig. 8 is constructed using the average number of edges between any two leaf nodes in two trees. Then the tree *ST* is constructed using these distances as shown in Fig. 9(a). This tree returned by the algorithm as supertree after step 4 because the set *DISTINCT* is empty. Fig. 9 also includes the results obtained by the Mincut and Adam's consensus method.

	A	*B*	*C*	*D*	*E*
A	0	2	2.5	3	2.5
B	2	0	2.5	3	2.5
C	2.5	2.5	0	2.5	3
D	3	3	2.5	0	2.5
E	2.5	2.5	3	2.5	0

Fig 8: Distance matrix obtained as a result of step 3 of DISTSUPERTEE.

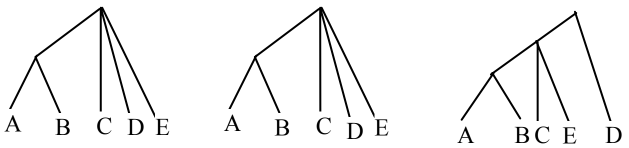
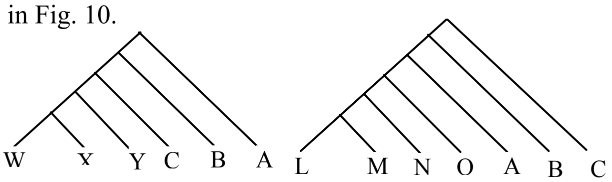


Fig. 9: Consensus trees as the result of combining tree of Fig.3 (a) DISTSUPERTREE (b) Adam's consensus method (c) Mincut.

DISTSUPERTREE gives the identical tree to the Adam's consensus tree when applied to the common leaf nodes, indicating that the DISTSUPERTREE method is at least as good as Adam's consensus method, which preserves all the nesting represented by all the input trees. The reason for the difference in output trees shown in Fig. 9 is that Mincut supertree method uses a local optimization principle and Adams consensus tree uses set theoretic operations. The disadvantage of the Mincut algorithm is that, the result of the algorithm sometimes contains the triplets which express conflict in input trees, an example of this is given below. Even though the DISTSUPERTREE uses an optimality criterion, it gives the similar results to Adams consensus tree. But the results are appreciable when applied to the tree representing the polytomy, which results when two or more lineages diverge from the single ancestor at the same time. An example is presented for the trees shown in Fig. 10.



(a) Tree *T'*. (b): Tree *T''*.

Fig 10: Trees *T'* and *T''* form an example of polytomy and are used for experiments.

In the trees, shown in Fig. 10(a) and (b), only three leaf nodes (A, B, and C) are common and shows a conflict on the relationship of these nodes. They also show resolved relationship for distinct leaf nodes. When trees shown in Fig. 10 are given as input to the DISTSUPERTREE the result of step 3 and step 4 is shown in Fig. 11.

	A	*B*	*C*
A	0	3	4
B	3	0	3
C	4	3	0

Fig. 11(a)

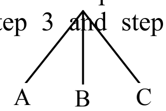


Fig. 11(b)

Fig. 11(a): Average distance matrix for COMM of Fig. 10(a) and 10(b), and 11(b): tree for Fig. 11(a).

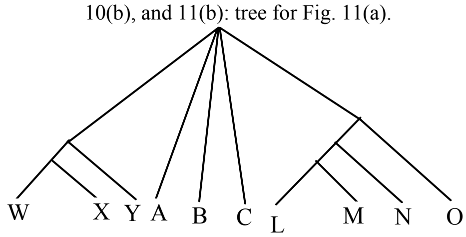


Fig. 12: The supertree constructed using DISTSUPERTREE for the tree shown in Fig. 10.

The subtrees of *DISTINCT* represent subtree of the corresponding tree without any common node. So the subtrees can be directly attached to the root node of the *ST*. The result of DISTSUPERTREE is shown in Fig. 12 and the result of Mincut is shown in Fig. 13.

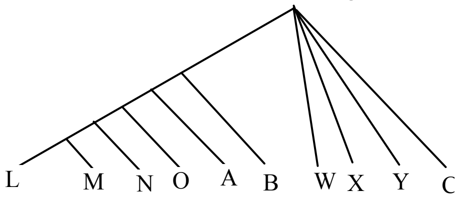


Fig. 13: The supertree constructed using Mincut for the tree shown in Fig. 10.

It is clear from the Fig. 12 and Fig. 13 that when the tree shown in Fig. 10(a) and 10(b) are given as input to DISTSUPERTREE and Mincut, the DISTSUPERTREE gives a better fit than Mincut. The input trees disagree on the relationship of A, B, and C, but the supertree computed using Mincut method contains triplet AB|C, thus resulting an unresolved supertree.. Both of these are local minima algorithms. There exists an exponentially large number of supertrees for a given set of tree and the above algorithm stops at local minima. The above example shows that the Mincut algorithm is sensitive to the size of the input trees and in addition, it fails to include the contradictory information of the input trees. There is no existing supertree or consensus method that displays all the uncontradicted information [1]. As it is always desirable to maximize the uncontradicted information that a supertree displays, our method fills the lacuna.

6. Conclusion

The DISTSUPERTREE algorithm, presented in this paper, demonstrates all desirable properties of supertree algorithms. It has a polynomial time complexity and results in a single output tree for a given set of input trees. Most of the supertree methods do not return a supertree for incompatible input trees. The supertree method presented in this paper always returns a single supertree. The supertree methods which depends on the Maximal Agreement such as Jesper et. al. [16], removes the edges which represents conflicting information results in removal of certain leaf nodes. MRP is a global optimization technique, which finds one or more optimal supertrees. But these methods are computationally heavy.

A Comparison of DISTSUPERTREE with different existing supertree methods is given in Table 1. The supertree methods are evaluated on the basic properties such as, time complexity, number of tree returned, results of the methods when the input trees carry evolutionary conflicts. Comparisons on polytomy, where more lineages diverge from the single ancestor at the same time, and optimization criteria are also made. The result of comparisons shows that DISTSUPERTREE out performs all the existing methods.

The algorithm DISTSUPERTREE can be further modified to work with more information given as input, such as ancestral time divergence and the input trees with some internal nodes labeled, than just rooted input trees.

Table 1: A comparison of different supertree methods

Algorithm	**Time Complexity**	**Number of trees returned**	**Result of incompatible input trees**	**Polytomy in input trees the method returns**	**Optimization technique**
MRP [2][3]	Exponential	Many	Many	Many trees	Global minima
Mincut [5]	Polynomial	One	One tree	Unresolved tree	Local minima
Modified Mincut [6]	Polynomial	One	One tree	Resolved tree	Local minima
RankedTree [9]	Polynomial	One	No tree	No tree	Local minima
Strict [20]	Polynomial	One	No tree	No tree	Local minima
DISTSUPERTREE	Polynomial	One	One tree	Highly resolved tree	Local minima

7. References

[1] M. Steel, A. Dress, S. Bocker. Simple but fundamental limitations on supertree and consensus tree methods, Syst. Biol, 49: 363-368. 2000.

[2] B. R. Baum. Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees, Taxon 41:3–10. 1992.

[3] M. A. Ragan. Phylogenetic inference based on matrix representation of trees, Molecular Phylogenetics and Evolution, 1:53–58. 1992.

[4] R.L. Graham, L. R. Foulds. Unlikelihood that minimal phylogenies for a realistic biological study can be constructed in reasonable computational time, Math. Biosci., 60:133-142. 1982.

[5] C. Semple, M. Steel. A supertree method for rooted trees, Discrete Applied Mathematics 105:147–158. 2000.

[6] R. D. M. Page. Modified mincut supertrees. In proceedings of the Algorithms in Bioinformatics, Second International Workshop, WABI 2002, Rome, Italy, pages 537–552.2002.

[7] D. Bryant C. Semple, M. Steel. Supertree methods for ancestral divergence dates and other applications. In English. O .R. P. Bininda-Emonds, editor, in phylogenetic supertree supertrees: combining the information to reveal the tree of life, pages 129-150, Kluwer Academics. 2004.

[8] R. D. M. Page and E. C. Holmes, Molecular evolution: a phylogenetic approach, Blackwell Science. 1998.

[9] M.A.H Zahid, A. Mittal, R.C. Joshi. A Heuristic Algorithm for Optimal Agreement Supertrees. In proceedings of the International Conference on Systemics, Cybernetics and Informatics (ICSCI-2005), Hyderabad, India, pages: 595-599, 2005.

[10] R. R. Sokal, C.D. Michener. A statistical method for evaluating systematic relationships, Univ. Kansas Sci. Bull., 28:1409-1438, 1958.

[11] N. Saitou, M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees, Molecular Biology Evolution, 4:406-425. 1987.

[12] M. Steel. The complexity of reconstructing trees from qualitative characters and subtrees, J.Classif. 9(1): 91-116. 1992.

[13] O. R. P. Bininda-Emonds, J. L. Gittleman, M. Steel. The (super) tree of life: procedures, problems, and prospects, Annual Review of Ecology and Systematics, 33:265–289. 2002.

[14] M. P. Ng, N. C. Wormald. Reconstruction of rooted trees from subtrees, Disc. Appl. Math., 69:19-31. 1996.

[15] B. Schieber, U. Vishkin. On finding lowest common ancestors: simplifications and parallelization, SIAM Journal of Computing, 17:1253-1262, 1988.

[16] 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.

[17] E. Adams. N-trees as nesting: Complexity, similarity and consensus, J. Classif. 3:299-317, 1986.

[18] A. D. Gordon. Consensus supertrees: the synthesis of rooted trees containing overlapping sets of labeled leafs, J. Classif., 3:31-39, 1986.

[19] D. Bryant. A classification of Consensus methods for Phylogenetics, DIMACS Series in Discrete mathematics and theoretical computer science. 2004.

[20] D.Bryant and M. Steel. Extension operations on leaf labeled trees, *Adv. Appl. Math.* 16:425-453, 1995.

[21] S. J. Willson, Constructing rooted supertrees using distances, Bulletin of Mathematical Biology, 66: 1755-1783, 2004.

[22] M.A, Bender, M. F. Colton, The LCA Problem Revisited, Proceedings of the 4th Latin American Symposium on Theoretical Informatics, 88-94, 2000.

[23] D. Harel and R. E. Tarjan. Fast algorithms for finding nearest common ancestors. *SIAM J. Comput.*, 13(2):338–355, 1984.