A SUPERTREE METHOD FOR COMBINING ROOTED PHYLOGENETIC TREES WITH ANCESTRAL DIVERGENCE TIME

M. A. H. Zahid, Ankush Mittal and R. C. Joshi

Department of E&CE, Indian Institute of Technology, Roorkee, Uttaranchal, India

{zaheddec, ankumfec, rcjosfec}@iitr.ernet.in

Abstract: Many supertree methods have been developed for amalgamating the small rooted phylogenetic trees with overlapping taxa into a single tree. Almost all the supertree methods combine the input trees based on the topological information carried by each of the input tree. Other evolutionary information such as fossil data, molecular dating data and actual divergence time estimates are usually ignored. If the available evolutionary information is considered with tree topology for the amalgamating the input collection of trees then the resulting supertree may be more accurate and resolved than the supertree constructed without using the additional information. The existing supertree methods with the ability to include additional evolutionary information are the extensions of the BUILD algorithm which has the property all-or-nothing, i.e., if the input collection of trees is incompatible, the algorithm fails to return the supertree. In this paper we propose a supertree method which incorporates the available evolutionary information and returns a supertree even for the incompatible collection of input trees. We modify the Adams consensus method to incorporate the additional evolutionary information and present the result of applying the algorithm to a hypothetical test data.

Introduction

Phylogeny, a tree of life of the lineages on the earth, provides a framework to facilitate biological information retrieval and predication. Currently most of the individual researchers and 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 the result of combining many smaller phylogenetic trees into a single tree. The smaller trees are combined with the constraint that the resulting supertree should represent the branching information, which includes the child parent and most recent common ancestor information, carried by each smaller input tree. Phylogenetic supertree construction continues to be one of most active field of research in systematics and classification, because only the supertree construction methods gives the hope of constructing and visualising the Tree of Life.

There exist many methods of combining smaller phylogenetic trees into a more comprehensive tree, which represent all the trees. If the input trees classify the same set of taxa or species then the resulting tree is called the consensus tree and if the input trees classify the overlapping set of taxa then the resulting tree is called a supertree. For a comprehensive survey on supertree methods refer [1].

Most of the existing supertree construction methods combine the input collection of trees based on the topological information carried by each tree. Other available evolutionary information such divergence data and molecular dating is usually neglected. This information can be used with topological information to result in a much more accurate and resolved tree. The additional information can be used after the supertree construction to resolve polytomies. Node that gives rise to three or more descendent lineages is called polytomy. This is the result of insufficient or conflicting information in the input trees. The resolution of polytomies in supertrees using divergence date information is an important step towards constructing more accurate and resolved Tree of Life.

Till date only RANKEDTREE [3] is the published algorithm which incorporates the additional information, such as divergence date, with topology for supertree construction. This is an extension of BUILD [4] and suffers with the problem of incompatible input trees. The disadvantages of the RANKEDTREE are as follows.

1. Cannot process the incompatible input trees
2. Do not represent all the nestings present in the input collection [2].
3. Assumes only topological incompatibility and neglects other conflicts, such as conflict in divergence dates information.

The definitions of different conflicts and other terminologies are given in Background and Preliminaries section.

In this paper we give a generalised algorithm based on Adams consensus tree [5] method, which overcomes the drawbacks of the RANKEDTREE. The proposed algorithm removes the divergence data conflict based on topological information. Moreover the algorithm also satisfies the desirable properties of the supertree construction methods as given in [6]. They are as follows

1. The method runs in polynomial time.

2. If input trees are compatible, the resulting supertree preserves the branching information carried by each tree. If more than one such tree exists then any one of them is produced.
3. The output is independent of the order, and re-labeling of leaf nodes of the input trees.
4. All the leaf nodes that occur in the input trees occur in supertree.

In this paper we propose a supertree method to incorporate relative time divergence information with tree topologies. This method returns a tree even for incompatible input trees, divergence dates conflicts and incompatibility between topology and divergence dates. We follow the least error method for the removing the conflicts. Once all conflicts are resolved, the Adam's consensus algorithm [5] is used to construct the supertree. Finally a rank function is used to rank the internal nodes of the resulting supertree based on the divergence date information. The algorithm is applied to a data set consists of hypothetical tress with divergence time.

In the following sections we discuss the background and preliminaries, algorithm with an example followed by the conclusions.

Background and Preliminaries

The terminology used in this paper is analogous to [6]. In this section we present some necessary and sufficient concepts to understand the problem and its solution.

A rooted phylogenetic tree, on label set *S*, is a tree *T* in which all the internal nodes have degree three or more, root node and leaf node have degree at least two and one respectively.

Let *T* be the rooted phylogenetic tree with the label set *S*. Given the label set *S'*, such that $S' \subseteq S$, the topological restriction of *T* to *S'* is the tree obtained by deleting the nodes which are not in the path from root to any node in *S'* and then contracting the internal edges whose degree two. The topological restriction is represented as $T'=T|_{S'}$. An example is shown in Figure 1. *T'* is called the induced subtree of *T* by *S'*. A rooted tree *T* is said to display another rooted tree *T'* if $S' \subseteq S$ and $T|_{S'}$ is isomorphic to *T'*.

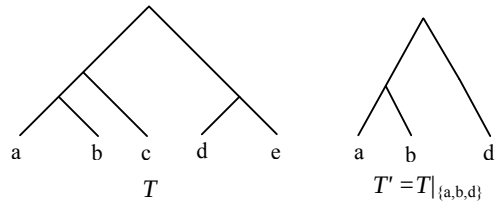

Figure 1: Two phylogenetic rooted trees. *T'* is induced subtree or restriction of *T* on the labels {a,b,d}.

Let *T* be a phylogenetic tree. For all $v_1, v_2 \in V(T)$, $v_1 \leq_T v_2$ represents that the node v_2 is descendent of v_1. The unique vertex of *T* that is the greatest lower bond of v_1 and v_2 under $\leq_T$ is referred to as the *most common recent ancestor* of v_1 and v_2 of *T*, represented as $MRC_T(v_1, v_2)$.

The three types of conflicts in the input data can be identified as follows:

- Conflicts in the input tree topologies
- Conflicts in divergence rates
- Conflicts between supertree topology and divergence date.

A set of phylogenetic rooted trees *G* is said to be topologically compatible if there exists a phylogenetic tree *T*, which display every tree in *G*.

The relative divergence date is a represented as "*div*(*x*,*y*) predates *div*(*u*,*v*)", which means that the divergence of the *x* and *y* predates the divergence of *u* and *v* species. A rank function, *R*, maps the interior nodes (*V'*) of the tree to some positive integer such that $\forall (v_0, v_1) \in V', R(v_1) < R(v_2)$ if v_2 is proper descendent of v_1. Let *X*, be a set of divergence dates, then it is said to be compatible, if for each data items in *X*, let the dating information on non-overlapping set of taxa *U* and *V* is given as "*div*(*U*) predates *div*(*V* "), then for all subsets *u* and *v* of *U* and *V* the respectively, then the divergence data should be "*div*(*U*) predates *div*(*v*)", $\{\forall u \subseteq U$ and $\forall v \subseteq V\}$,otherwise there exists a divergence date conflict.

Similarly conflict may occur between topology and divergence date information. For example, consider the supertree topology given in figure 2, and divergence date information is given as "*div*(*c*,*d*) predates *div*(*d*,*f*)". Both the tree and divergence date represent the contradicting information.

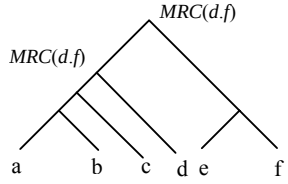

Figure 2: Tree topology contradicting the divergence date, "*div*(*c*,*d*) predates *div*(*d*,*f*)".

We resolve the conflict between the topology and divergence date information by giving the priority to the topological information carried by the tree and neglecting the divergence date information.

The DdateSupertree Algorithm

In this section we propose the DdateSupertree algorithm, which incorporates the divergence date information with topological information. We also establish some of the properties and conflict detection and resolution techniques used in algorithm. We formulate a necessary and sufficient condition for detection of conflict in the divergence date information and then proposed and algorithm to resolve it.

Lemma 1, 2 and 3 provide the necessary and sufficient conditions for the detection of conflict in divergence data, and the theorem 1 gives the conflict resolution results.

Lemma 1: Let T is a tree with label set L (T), which satisfies the divergence date information "*div*(X) predates *div*(Y)", where $X, Y \in L(T)$. Let x and $y$are subsets of X and Y, $x \subseteq X$ and $y \subseteq Y$, respectively then all of the following conditions should hold for compatibility.

1. *div*(X) predates *div*(x) or $MRC_T(X) = MRC_T(x)$
2. *div*(Y) predates *div*(y) or $MRC_T(Y) = MRC_T(y)$
3. *div*(X) predates *div*(y)

Proof: The tree shown in figure 3 is satisfies the predate condition,"*div*(X) predates *div*(Y)", where $X, Y \in L(T)$. Now it can be easily proved that the three conditions given in lemma should hold for compatibility.

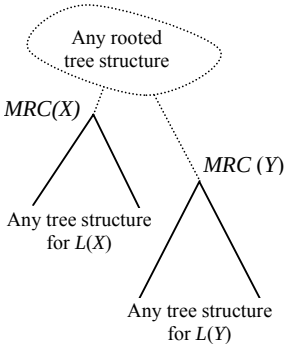

Figure 3: A generalized tree satisfying "div(X) predates div(Y)" divergence date information.

To prove first two conditions it is enough to prove that the *MRC*(N) will be the root of the restricted tree $T|_N$, where $N \subseteq L(T)$, as node will have the least rank. Let T be a tree with label set L(T), its restriction on a set of labels, N, where $N \subseteq L(T)$, can be obtained by removing all the leaf nodes which are not in N and suppressing the nodes with degree two. Obviously the resulting tree is $T|_N$ with N as set of leaf nodes and *MRC*(N) as root. Therefore any subtree of T' of T have root rank greater than the rank of the root of the parent tree T. This proves first two conditions.

The ranks for most recent common ancestor of X and Y, in relative representation is $R(X) < R(Y)$. The value assigned by rank function to the most common recent ancestor of Y is always greater than the rank of most recent common ancestor of X, as *div*(X) predates *div*(Y). If we take a set y, $y \subseteq Y$, then the *MRC*(y) may be a descendent of the *MRC*(Y) or is same as *MRC*(y) and its rank function value will be $R(Y) \le R(y)$. We know that $R(X) < R(Y)$ or *MRC*(X) predates *MRC*(Y), therefore the value of R(y) should be greater than R(X). the divergence relation between X and y can be given as $R(X) < R(Y) \le R(y)$. This proves third conditions and hence the lemma. □

Lemma 2: Let the divergence date information "*div*(X) predates *div*(Y)", where $X, Y \in L(T)$, is given. If there exist a tree T, with L(T) labels. If *MRC*(X)=*MRC*(Y) then the divergence information, which includes X and Y on different sides of divergence notation is incompatible.

Lemma 3: Let T is a tree with label set L(T), which satisfies the divergence date information "*div*(X) predates *div*(Y)", where $X, Y \in L(T)$. If Z is set with the property $Z \cap X \neq \phi$ and $Z \cap Y \neq \phi$, than "*div*(Z) predates *div*(X)and *div*(Y)".

We use the results of the lemma 1,2 and 3 for the detection of conflict in the divergence date information. Once the conflicts are identified the divergence date statements with maximum number of conflicts are removed. The conflicts in the tree topology and divergence data is removed by giving the priority to the tree topology as it is a result of amalgamating many smaller trees and displays all the trees. The algorithm for finding the conflict is DivCompat is given below.

Algorithm: DivCompat (D **)**
Input: divergence date data D. The divergence date is given in "*div*(X) predates *div*(Y)" form.
Output: compatible divergence data.

```
begin
  sort D by the number of labels in it , in descending
  order and attach a conflict flag to each statement .
repeat until all statements have conflict values zero
initialize all conflicts to 0 (zero).
begin
  for each statement, s, in sorted D
    for s+1 to last statement of D
      if there is a statement on X and y, y ⊆ Y then
          The statement should be "div(X) predates
          div(y)".
      else
          increment the conflict by one.
      end if
      if there is a statement on X and Z, Z ∩ X ≠ φ then
          The statement should be "div(Z) predates
          div(X).
      else
          increment the conflict by one.
      end if
    end for // first for
  end for // second for
  for each statement in D
    remove the statement with maximum conflict and
    modify the D.
  end for
 end repeat
 retrun D
end (Algorithm)
```

The DivCompat algorithm takes the divergence date information as input and returns compatible divergence information. The theorem 4 gives the result obtained by the algorithm.

Theorem 4: Let the divergence date information D is applied to the algorithm DivCompat results in a compatible set of predate information statements.

Proof: proof for this is obvious and can be established easily with using lemma 1, 2 and 3. □

The algorithm DdateSupertree makes use of DivCompat and Adams consensus tree construction algorithm to result in a ranked supertree. The algorithm is as follows.

Algorithm: DdateSupertree (*D*, *G* **)**
Input: divergence date data *D*. all the divergence data is given in "*div*(*X*) predates *div*(*Y*)" form. *G* is collection of, *k*, input trees.
Output: ranked supertree according to divergence date information *D*.
begin
$X \leftarrow \phi$
D = DivCompat(D)
$X = X \bigcap_{i=1}^{k} (T_i)$
CT = AdamsTree($T_1|_X, T_2|_X, ..., T_k|_X$)
for *i*=1 to *k*
for each label *l*, $l \in (L(T_i) - X)$
add a new edge between an edges of *CT* and *l* in such a way that there should not be any topological conflict. Modified tree is *ST*.
end for
end for
for each statement *s* in *D*
assign ranks to the *MRC*(*X*) and *MRC*(*Y*) according to the statements and modify the *ST* if necessary. Modified tree is *RT*.
end for
return *RT*
end (ALGO)

To illustrate the DdateSupertree, it is applied to the tree shown in figure 4 as input collection of trees and divergence date information is, "*div*(*c*,*d*) predates *div*(*a*,*b*)", and "*div*(*a*,*e*) predates *div*(*c*,*d*)". The final result of the algorithm is shown in figure 6.

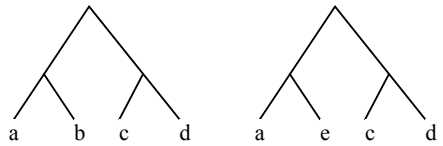

Figure 4: input trees for DdateSupertree.

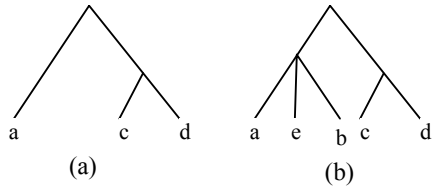

Figure 5: (a) Restriction on {a,b,c} and
(b) Supertree of input trees shown in figure 5.

The two trees shown in figure 4 classify overlapping set of taxa with *a* ,*b* and *c* are common to both the trees. The restriction of both the trees on *a*, *b* and *c* is sameand is shown in figure 5(a). The supertree for the input collection before ranking is shown in figure 5(b).

The final supertree after incorporating the divergence information given is shown in figure 6.

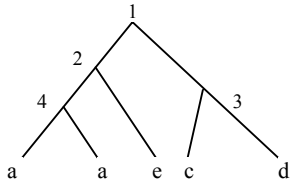

Figure 6: Final ranked supertree after incorporating divergence information.

Conclusions

In this paper we proposed an algorithm for incorporating additional information into the supertree algorithm, which is useful in resolving polytomies and results in more accurate and refined tree. Till date only one algorithm, RankedTree [3], exits with the capability to incorporate divergence date information for supertree construction. It is an extension of BUILD [4], which could not process incompatible collection of input trees. Moreover, the supertree constructed with [4] or its variants may not represent the nesting present in all the input trees [2]. In contrast, DdateSupertree, is an extension of Adams consensus tree, and thus DdateSupertree represents all the nesting represented by the all the input tree [5].

We identified and proposed sensible method to resolve additional conflicts such as, conflict in divergence date information, conflict between tree topology and divergence information.

References

[1] Bininda-emonds, O. (2004): 'The evolution of supertrees', *Trends in Ecol. and Evol.*, **19**, pp. 315-322.

[2] Denial, P., Semple, C. (2004): 'A class of general supertree methods for nested taxa', *SIAM Journal of Discrete Maths*, in press.

[3] Bryant, D., Semple, C., Steel, M., (2004): 'supertree methods for ancestral time divergence date and other applications', in Bininda-emonds, O. (Ed): 'Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life', Computational Biology Series, Kluwer, pp. 129-150.

[4] Aho, A.V., Yehoshua, S., Szymanski, G. T., Ullman, J.D. (1981): ' inferring trees from lowest common ancestors with an application to the optimization of relational expressions', *SIAM Journal Comput.*, **10**, pp. 405-421.

[5] Adams, E.N., (1972): Consensus techniques and comparison of texanomic trees', Sys. Zool., **21**, pp. 390-397.

[6] Semple, C., Steel, M., (2000): 'A supertree method for rooted trees', *Disc. Appl. Math.*, **105**, pp. 147-158.