

# Calculation, Visualization, and Manipulation of MASTs (Maximum Agreement Subtrees)

Shiming Dong and Eileen Kraemer  
Computer Science Department,  
The University of Georgia  
{dong, eileen}@cs.uga.edu

## Abstract

*Phylogenetic trees are used to represent the evolutionary history of a set of species. Comparison of multiple phylogenetic trees can help researchers find the common classification of a tree group, compare tree construction inferences or obtain distances between trees. We present TreeAnalyzer, a freely available package for phylogenetic tree comparison. A MAST (Maximum Agreement Subtree) algorithm is implemented to compare the trees. Additional features of this software include tree comparison, visualization, manipulation, labeling, and printing.*

### **Availability:**

<http://www.cs.uga.edu/~eileen/TreeAnalyzer>

**Keywords:** phylogenetic tree, MAST, metric

## 1. Introduction

A *phylogenetic tree* is a structure frequently used to represent the evolutionary history of a set of species. According to the formal definition in [1], a phylogenetic tree (or an *evolutionary tree*) for a species set  $A$  is a rooted tree in which the leaves (or taxa) are uniquely labeled by the species in  $A$ , and the internal vertices represent ancestors. In a rooted phylogenetic tree, the root is the ancestor of all the nodes that compose the tree. An unrooted phylogenetic tree, in which no interior vertex has only two incident edges [2], does not allow the determination of ancestors and descendants. In either case, the branch length usually represents the number of changes that have occurred in that branch and is proportional to time or to the number of fixed novel mutations that have accumulated.

Constructing a phylogenetic tree is a fundamental problem in the field of bioinformatics, explicitly providing information about evolutionary relationships

among the species represented in the trees, and further providing insight into questions about topics such as the biochemical machinery of the organisms [3]. Studies of evolutionary trees are also helpful for research in gene evolution, population subdivision, analysis of mating systems and paternity testing, as well as studies of individual relatedness, geographic variation, and species boundaries.

Constructing a phylogenetic tree is an estimation procedure [4]. Generally, we do not have direct information about the past and can only access contemporary data. Thus, the procedure of constructing a phylogenetic tree is a “best estimate” of an evolutionary history, made based on incomplete information contained in contemporary data. The phylogenetic tree construction methods that have been developed are based on various criteria such as the parsimony criterion, the distance criterion and the maximum likelihood criterion [4]. The “perfect” phylogenetic tree is the one that reflects the true evolutionary relationships among species in the real world. Phylogenetic relationships inferred from a specific set of sequences should, in theory, be congruent if the sequences have the same overall history [4]. However, different tree construction methods based on different criteria with the same set of species may result in different trees. No method exists that is ideal for all performance criteria [5]. Thus, phylogenetic tree comparison, which is able to provide similarity and dissimilarity information for the tree structures, has been investigated in depth in the field of bioinformatics. A number of tree comparison metrics have been proposed. Examples include the partition metric [6, 7], the quartet metric [8, 2], the NNI metric [9], the consensus tree metric [10] and the MAST (Maximum Agreement Subtree) metric [11]. In our project, we address the use of the MAST metric for tree comparison.

MAST (Maximum Agreement Subtree) is a method that combines information from rival phylogenetic

trees into a new agreement subtree. According to [11], MAST may be defined as follows: "Given two binary trees, a largest subtree contained in both of the original trees that has been obtained by pruning vertices is called an agreement subtree. A maximum agreement subtree is an agreement subtree resulting from pruning the fewest number of end-vertices from the two trees." The MAST approach is useful for several reasons [12]: (1) It can provide increased confidence in the quality (optimality or near-optimality) of a specific phylogenetic tree. (2) It can summarize the relationships common to multiple alternative trees. (3) It can be used to investigate the stability of a tree topology. UMAST is the unrooted version of MAST.

The MAST metric is used both to present the common substructures in the compared trees and to identify areas of conflict in the trees. Compared to the consensus tree metric, MAST provides more precise information on the similarity of the trees. The MAST metric prunes unstable leaves, a consideration ignored by the consensus metric. "One problem with many consensus techniques is that the consensus tree can contain information that is not present in all of the input trees, and sometimes not in any of the trees." [13] The resultant trees from the MAST metric show the common relationships among a variety of trees on the same given set of data. This approach is particularly useful when only a few taxa are responsible for the incongruity among trees, thereby providing a means of identifying "unstable" taxa. Drawbacks of the MAST metric are the difficulty and computational requirements for identifying the maximum agreement subtree [14].

We have developed **TreeAnalyzer**, a phylogenetic tree visualization and analysis software developed to meet the needs of research microbiologists at the Russell Research Center of the U.S. Department of Agriculture(USDA)<sup>1</sup>. TreeAnalyzer reads a phylogenetic tree from a NEXUS [15] or TreeAnalyzer format file and visualizes it in a tree structure. The MAST metric is used to compare multiple trees and the resultant MASTs are visualized by mapping them onto the original compared trees. The highlighted MASTs are able to directly differentiate the common area from the dissimilar area of the compared trees. A normalized number is computed to represent the proportion of common parts of those trees. The compared trees may have different numbers of leaves or different leaf names. Theoretically, there is no limitation on the number of trees compared and the number of leaves they have.

---

<sup>1</sup> We wish to acknowledge the many contributions of Dr. Greg Siragusa and Dr. Kelli Hiatt in shaping the goals and requirements of this software.

Good performance has been observed for sample trees of real data with fewer than 120 leaves. In practice, the target users typically deal with trees containing up to 50 to 100 leaves. Users may interact with the visualized phylogenetic trees and the created MASTs to perform manipulations such as modifying leaf names, swapping the subtrees of a specific internal node, or changing the font of leaf labels. This software is developed using Java [16] and runs on any platform that supports Java.

The goal of the microbiologists in using this software is to compare the genotypic traits of different sets of bacterial isolates in order to determine either a common source ("source tracking") or a common phylogenetic lineage ("clonality") of individual representative bacterial isolates. To accomplish this, microbiologists use many different subtyping methods. Some methods are sequence-derived, while others are restriction site (banding pattern) based. The users would like the software to compare the trees obtained by these two different methods to determine how well the results of these approaches correspond. Many algorithms to implement the MAST metric have been proposed and we choose the algorithm introduced by Farach et al. in [17] to do multiple tree comparison and the algorithm presented by W. Goddard et al. in [18] to compare two trees. Both of these algorithms provide exact solutions, their speeds are acceptable, and the algorithms are relatively straightforward to implement.

## 2. TreeAnalyzer's Usage and Features

The input file format for TreeAnalyzer is the NEXUS format [15], which is used by a number of popular phylogeny programs such as PAUP [19], Phylip [20] and MacClade. The NEXUS format file consists of the "#NEXUS" directive at the beginning of the file, the translation table and the tree blocks. The translation table starts with the label "TRANSLATE". It is used to map the sequence names in the real world into the internal leaf labels. Figure 1 depicts a sample NEXUS file. The numbers on the left side "1", "2", ..., "7" are the internal leaf labels, and the strings of "K@riboprint1@00000002", etc. are the real world sequence names. To simplify the representation of the tree structure, the internal leaf labels, instead of sequence names, are used to describe the tree topology. Trees are described using a standard parenthetical notation. Each cluster in the tree is enclosed by a pair of parentheses "( )". In such a cluster (1:3.55, 2:3.55), the string to the left of the colon is a node's leaf label and the floating point number to the right of the colon is the branch length between that node and its ancestor. In the example shown, each line that starts with

“UTREE PAUP” is a tree topology. A detailed definition of the NEXUS format is provided in [15].

```
#NEXUS
BEGIN TREES;
  TRANSLATE
    1   K@riboprint1@00000002,
    2   K@riboprint1@00000001,
    3   K@riboprint1@00000005,
    4   K@riboprint1@00000007,
    5   K@riboprint1@00000006,
    6   K@riboprint1@00000003,
    7   K@riboprint1@00000009,

  UTREE PAUP_1 =
  (((((1:3.55,2:3.55):2.89,3:6.44):6.03,(4:5.73,5:5.73):6.73)
  :1.95,(6:0.35,7:0.35):3.7);
  UTREE PAUP_2 =
  (((((7:1.3,3:1.3):2.4,(5:0.9,4:0.9):1.5):1.3,2:2.3):3.5,(1:0.
  8,6:0.8):1.4);
  ....
  UTREE PAUP_n =
  (7:0.9,((3:1.6,(5:3.4,4:3.4):1.5):1.4,(2:0.9,(1:0.4,6:0.4):1.
  7):1.7):1.6);

ENDBLOCK;
```

**Figure 1. Nexus format sample**

## 2.1. Functions

Through the TreeAnalyzer user interface (Figure 7) users may perform file operations (open, close, save and print) and phylogenetic tree operations (subtree swap, label change, MAST computation, label head change). To view a phylogenetic tree or perform operations, the user must first read the binary tree topologies and leaf sets from files. The tree structures labeled with branch lengths are then displayed as seen on the left side of the panel. Information about the corresponding leaf names, ribotypes and comments are provided as seen on the right side of the panel. An image available for riboprints can also be loaded beside the corresponding leaf, seen in the center panel. Each tree’s display is accessible on a separate tabbed pane.

Users may interact with the visualized trees to swap subtrees, change leaf names, modify header names and change the color and fonts of leaf names. To do so, the user first chooses a mode by clicking on an icon in the toolbar. The icons, from left to right, support rotation at nodes, editing of labels, editing of headers, creation of a MAST, and printing. Rotation at nodes permits the user to transform a tree into any isomorphic tree. Users may wish to apply such a transformation in order to provide a more logical ordering of leaves. Created or modified trees can be saved as an object. The print

function permits the trees to be printed out or sent to image files, such as tiff and bmp format files, which may then be included in papers, posters, slides, etc.

As seen in figure 8, MASTs are visualized as highlighted areas on a pair of trees selected from the compared tree group, using both color and line thickness. Users may select other tree pairs for visualization. Corresponding leaves are connected by red lines; a “matching” procedure is applied to minimize line crossings. The tree distance and the *similarity index* are computed and displayed. The similarity index is a metric  $x/y$ , where  $x$  is the proportion of leaves that remain in the MAST from the original leaf set and  $y$  is the proportion of leaves pruned to produce the MAST. The MAST shows a direct picture of which parts of the compared trees are common, and the similarity index provides a quantitative measure of how similar or dissimilar the compared trees are.

The users may perform the same operation on MASTs as on the original trees: swapping the subtrees of a specific node or changing the fonts and color of leaves. MASTs may be saved or printed. MASTs are displayed one at a time. Trees may be loaded from multiple NEXUS files or a tree group may be loaded “en masse” from a single NEXUS file. The software supports multiple tree comparison among unrooted binary trees in which differences exist in the leaf sets. We use the algorithm described in [18] for two tree comparison and that introduced in [17] for multiple tree comparison.

## 3. TreeAnalyzer Algorithms

The algorithm described in [17] by Farach, et al., which is fast and provides an exact solution, is implemented for multiple tree comparison. Before the comparison procedure, all the trees are checked and those leaves that do not belong to the common leaf set of the tree group are pruned. The two-tree comparison uses the algorithm introduced by Goddard et al. in [18]. That algorithm is only applied for two-tree comparison but is faster than the Farach algorithm. It also produces MAST tree structures in addition to the MAST leaf set, which are useful for computing the specific tree distance in [18].

### 3.1. MAST Algorithm Background

During the past twenty years, many researchers have been working on the MAST problem. Finden and Gordon proposed a pruning algorithm in 1985 [12]. This is an approximation algorithm, with time complexity of  $O(n^5)$  where  $n$  is the number of leaves in the trees. In 1992, Kubicka and Kubicki provided an

exact algorithm with time complexity  $O(n^{\log n})$  [11]. The SW(Steel and Warnow) algorithm [21] is an  $O(n^2)$  algorithm for the MAST problem. It is based on dynamic programming. Since 1993, several algorithms have been developed based on the SW algorithm. Farach and Thorup in their paper [22] suggest an efficient method to simplify the SW algorithm. They derived an algorithm with time complexity of  $O(n^2)$  for the binary rooted tree, and  $O(n^2 c \sqrt{\log n})$  for the unbounded unrooted case, where  $c$  is a constant. In 1995, Farach and Thorup developed an algorithm that improved the time complexity of the previous polynomial algorithms. They gave an  $O(n \log^3 n)$  time algorithm for MASTs for the binary trees [1]. The fastest algorithm thus far is that provided by Cole et al. in [23]. That algorithm is based on Farach and Thorup's work and makes some improvements by presenting an  $O(n \log n)$  algorithm.

### 3.2. Algorithms for TreeAnalyzer

**3.2.1. Algorithm for two-tree Comparison.** Two-tree comparison in TreeAnalyzer is based on the MAST algorithm presented by Goddard et al. in 1993, which takes  $O(n^2)$  time [18]. Although it is not the algorithm with the smallest time complexity, its speed is acceptable, it produces an exact answer, and its implementation is straightforward. They provide an exact polynomial-time algorithm for comparing two trees, based on dynamic programming, for rooted trees, and then generalize the algorithm to unrooted trees. In addition, the authors present another algorithm to compute the tree distance - the sum of distances between the pruned leaves with respect to MAST. The tree distance provides accurate and detailed information on how dissimilar the two trees are. Furthermore, their algorithm allows the compared trees to be of different sizes.

The algorithm begins by assuming that  $T_a$  is a tree rooted at a vertex  $a$ , with children  $b$  and  $c$ , and that  $U_w$  is a tree rooted at  $w$ , with children  $x$  and  $y$ . The algorithm considers each of the possible matchings of these subtrees, and calculates the size of the agreement subtrees that would result if the subtrees were matched. Suppose tree  $T$  has size  $m$  and tree  $U$  has size  $n$ , a table with the size of  $(2m-1)*(2n-1)$  is built to store all of the intermediate *masts*. Each element  $C_{ij}$  ( $1 \leq i \leq m$ ,  $1 \leq j \leq n$ ) in the table represents *masts* between a specific subtree of  $T$  and a specific subtree of  $U$ . The table is filled from left to right and from top to bottom according to a post-fix order traversal of trees  $T$  and  $U$ . An initial condition is set as follows: when we try to get *masts* of  $T_a$  and  $U_w$ , if  $a$  is a leaf and if  $U_w$  also contains a leaf labeled  $a$ , then the MAST of  $T_a$  and  $U_w$  is  $a$ , otherwise the MAST of  $T_a$  and  $U_w$  is  $\emptyset$ ; if  $U_w$  is a

leaf, vice versa. If neither  $T_a$  nor  $U_w$  is a leaf, then we search the previously computed *masts* in the table and compute the specific *mast* according to the possible matchings. Thus, each element in the table can be computed through dynamic programming. Finally we get MAST for trees  $T$  and  $U$ , which is the  $(2m-2)*(2n-2)^{th}$  element in the table. This algorithm not only retains the MAST tree structures but also keeps the similarity information and the size of the MAST trees of the compared trees.

In order to compare two unrooted trees, we fix the root for the tree  $T$  as  $T'$  and try all the possible locations of root for the second tree  $U$ . The comparison between any possible rooted trees  $T'$  and  $U'$  uses the rooted tree comparison algorithm mentioned above. The final solution of  $T$  and  $U$  is the collection of all  $T'$  and  $U'$  pairs:  $UMAST = \max(\Sigma MAST(T', U'))$ .

For rooted trees  $T$  and  $U$ , a  $(2m-1)*(2n-1)$  table is constructed, where  $m$  is the size of the leaf set of  $T$  and  $n$  is the size of the leaf set of  $U$ . The time complexity is  $O(mn)$ . Generalizing the rooted version to the unrooted version requires moving the root of one tree  $O(n)$  times. The computation takes  $O(m)$  for new rooted  $T$  and  $U$ . Thus, the total time complexity is  $O(mn)$ . If tree  $T$  and  $U$  have the same number of leaves, it is  $O(n^2)$ .

**3.2.2. Algorithm for multi-tree comparison.** The algorithm in [18] provided a fast and optimal solution for the MAST problem of two tree comparison; however, its application on multiple tree comparison is not as efficient as that on two trees. In order to get the common tree structure for all the trees compared, almost all the intermediate results must be retained until the MAST size is computed. For a tree group with more than five trees, each of which has fifty leaves (a typical tree size in biology), this computation is too time-consuming. Even for the faster branch-and-bound algorithm, speed is still a problem. Thus for multiple tree comparison, we choose the MAST method from [17].

This algorithm is also based on a dynamic programming method. It is applied to the comparison of multiple  $d$ -degree trees. Since a *mast* set (the set of leaves in the *mast*) uniquely defines an agreement subtree given a set of trees, this algorithm aims to find the maximum *mast* set of the tree group instead of the *mast* tree topology. The *mast* topology for rooted tree group can then be computed within the time bounds  $O(n^3)$ , given that we deal only with binary trees. Details of the algorithm may be found in [17]. We implement this using a recursive method. The algorithm can also be generalized to the unrooted version by enumerating each leaf in the leaf set as a root for the tree group and choosing the agreement

subtrees with the maximum size, which takes time complexity of  $O(n^4)$ .

## 4. Use in practice

### 4.1. Sample run

As an example, we describe the application of our tool to a target problem by users at the USDA. The users want to compare two phylogenetic trees “Salmonella3repprimers” and “SalmonellaRiboprint”, constructed by different methods but based on the same leaf set. As seen in figure 9, the tree structure of “SalmonellaRiboprint” is displayed on the left side and the branch lengths are displayed. Leaf names are shown on the right side of the panel.

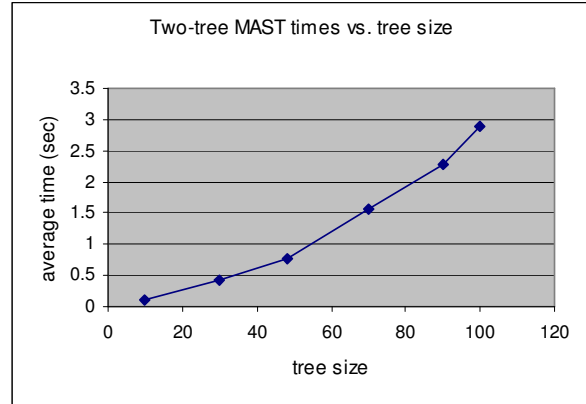
Leaves of particular interest are highlighted by using different colors. After computing the MAST, we get the results as shown in figure 8. From figure 8, we find that there are four MASTs created and all of them have the same size but different tree structures. The displayed MAST is highlighted by the blue thick lines. The corresponding leaves that belong to the MAST are connected by the red lines. We see from the display that the distance between “Salmonella3repprimers” and “SalmonellaRiboprint” is 33.102, and that both the compared tree are of size 50. This means that only a small fraction of the trees are common and 33 leaves must be pruned to produce the MASTs. Another numerical result, the similarity result, also supports this conclusion. Thirty-three percent of the leaves are retained in MASTs and sixty-six percent of the leaves are pruned, which demonstrates that these two trees are only moderately similar to one another. The time spent on the comparison of the two trees of 50 leaves is less than one second, fast enough for the purposes of these users.

### 4.2. Performance Numbers on Trees of Various Sizes

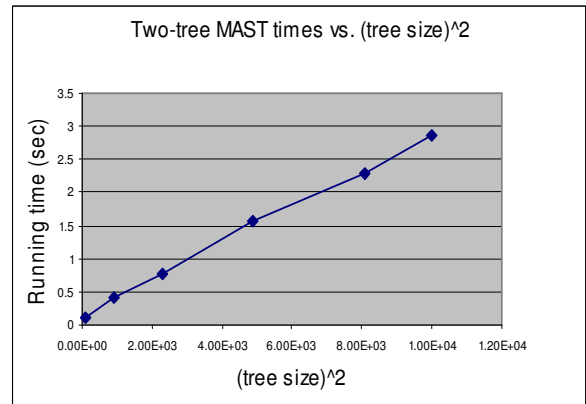
We used LumberJack [24], a phylogenetic inference tool, and sequence data files provided with the package to create groups of phylogenetic trees of various sizes. Each tree group is based on the same dataset and contains at least three trees. The results from two-tree groups are shown in table 1, figure 2 and figure 3, and the results from three-tree groups are shown in table 2, figure 4 and figure 5. We can verify from the plots that the two-tree algorithm exhibits  $O(n^2)$  behavior while the multi-tree algorithm exhibits  $O(n^4)$  behavior.

**Table 1. Average running time for MASTs on two-tree groups of indicated size**

Tree Size	10	30	48	70	90	100
Avg. time (sec)	0.10	0.41	0.76	1.57	2.28	2.88



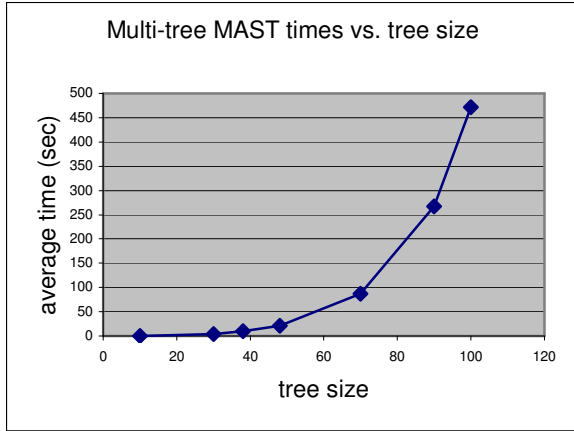
**Figure 2. Average running time for MASTs of two-tree groups, plotted versus tree size.**



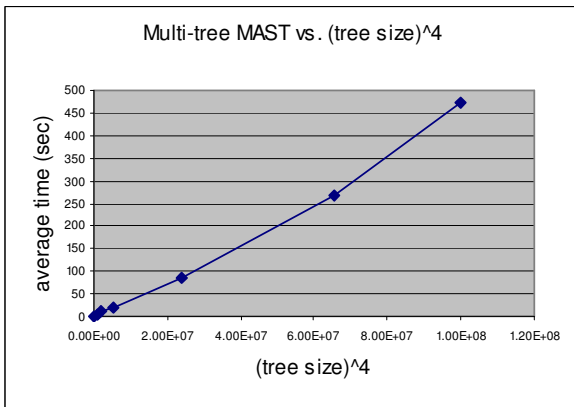
**Figure 3. Average running time for MASTs of two-tree groups, plotted versus (tree size)<sup>2</sup>, indicating  $O(n^2)$  behavior.**

**Table 2. Average running time for MASTs on three-tree groups of indicated size**

Tree Size	10	30	38	48	70	90	100
Avg. time (sec)	0.07	3.90	9.81	21.30	87.17	267.29	471.36



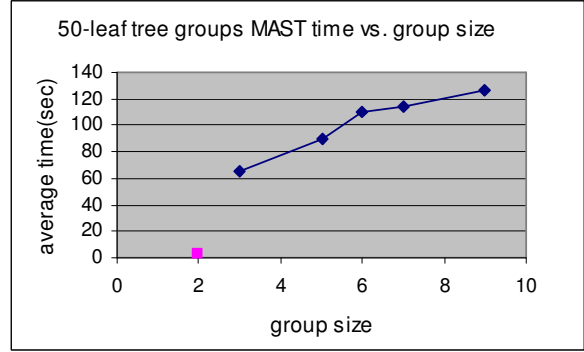
**Figure 4. Average running time for MASTs of three-tree groups, plotted versus tree size.**



**Figure 5. Average running time for MASTs of three-tree groups, plotted versus (tree size)<sup>4</sup>, indicating O(n<sup>4</sup>) behavior.**

**Table 3. Average running time for MASTs on three-tree groups of indicated size**

Tree Group Size	2	3	5	6	7	9
Avg. time (sec)	2.93	65.92	90.38	109.99	113.59	126.63



**Figure 6. Average running time for MASTs of 50-leaf tree groups, plotted versus group size.**

We then constructed 5 sets of 50-leaf trees. The sets were of size 2, 3, 5, 6, 7 and 9. MASTs were constructed for each group and times were as reported in table 3. The two-tree group used the Goddard algorithm [18], while the Farach algorithm [17] was applied to all other groups. It can be seen from figure 6 that running time varies linearly with group size in groups of size 3 or larger.

## 5. Related Work

Many applications of phylogenetic tree analysis and comparison have been developed, such as PAUP [19], Phylip [20] and REDCON [25]. Each of these packages has its own advantages and limitations. PAUP is one of the most widely used phylogenetic tree processing software packages. It provides multiple tree construction algorithms, tree manipulations and tree display methods. It works on Unix, Windows and Macintosh platforms. However, in the windows version of PAUP most of the operations are implemented through a command-line interface. PAUP does not support the printing or saving of trees in its Unix, windows or DOS version, which is an important feature for our users. Phylip is also a powerful software package for phylogenetic tree construction and manipulation. It can also be used to compute the tree distance and obtain the consensus trees among a tree group. However, it does not provide a user interface and all of the functions must be run through the DOS command line. Also, the users must run multiple different executable programs in the package in order to implement a series of operations, which is inconvenient. Ntsyspc [26] is a software package composed of two programs for phylogenetic tree text file editing, tree construction and manipulation. It provides the consensus metric instead of the MAST metric for tree comparison and neither the displayed phylogenetic tree nor the resulting consensus trees can

be manipulated or interacted with. COMPONENT [27] is a free phylogenetic tree processing software on the windows platform. It provides multiple tree comparison methods, such as the quartet metric, partition metric, consensus metric and the MAST metric. However, in the case of the existence of multiple equivalent MASTs, it produces only one MAST. The solution is incomplete and not representative. In addition, the algorithm used is slow, and it does not work for comparing trees with more than fifty leaves. REDCON, developed by Mark Wilkinson in the department of Zoology of the Natural History Museum, is a phylogenetic tree comparison software. It provides multiple consensus methods, such as the strict consensus metric and the majority consensus metric. However, it is a DOS command based software and it has a severe limitation on both the number of trees compared and the number of leaves the trees may contain. All of its tree comparison methods are unable to deal with trees with more than eighty leaves, which is not an unusual phylogenetic tree size. Mesquite [28] is a powerful software for evolutionary biology. It includes phylogenetic analyses and population genetics analyses. It also provides tree comparison metrics. However, it only provides the consensus metric to show the common structure within tree groups and a real value assigned to each taxon to represent the taxon instability among trees. MAST is not considered in that software.

## 6. Future Work and Conclusion

In this paper, we introduced TreeAnalyzer, a phylogenetic tree comparison and visualization tool. TreeAnalyzer helps researchers to determine common phylogenetic structure and a quantitative index of closeness of trees. The similarity of compared phylogenetic trees indicates the relationships between two sequences and can provide insight into how those organisms have evolved.

Compatibility is one of the features of TreeAnalyzer. TreeAnalyzer allows efficient access to the generally used PAUP NEXUS format of tree data and provides MAST output that can either be saved as an object file or exported to an image file.

Neither of the two algorithms implemented takes into account the branch lengths. Algorithms do exist for the maximum weighted agreement subtree problem [29], which uses branch length information. However, these require the exact integer distance between each node in all the trees, which is not usually known. If the exact branch length is unknown, the time complexity of the trees with approximated branch length is  $O(kn^{d+1} + n^{2d})$ , where  $k$  is a constant,  $n$  is the number of

leaves in the trees, and  $d$  is the degree bound of the trees [29].

The software has a sound user interface. It differs from other phylogenetic tree comparison software in several aspects. It reads input data and visualizes it with a tree structure that the users can interact with. MASTs are mapped on the original trees to highlight or emphasize the areas of the trees that differ. Users may interact with MASTs by swapping the subtrees of any internal node, modifying leaf names or replacing the trees as a view.

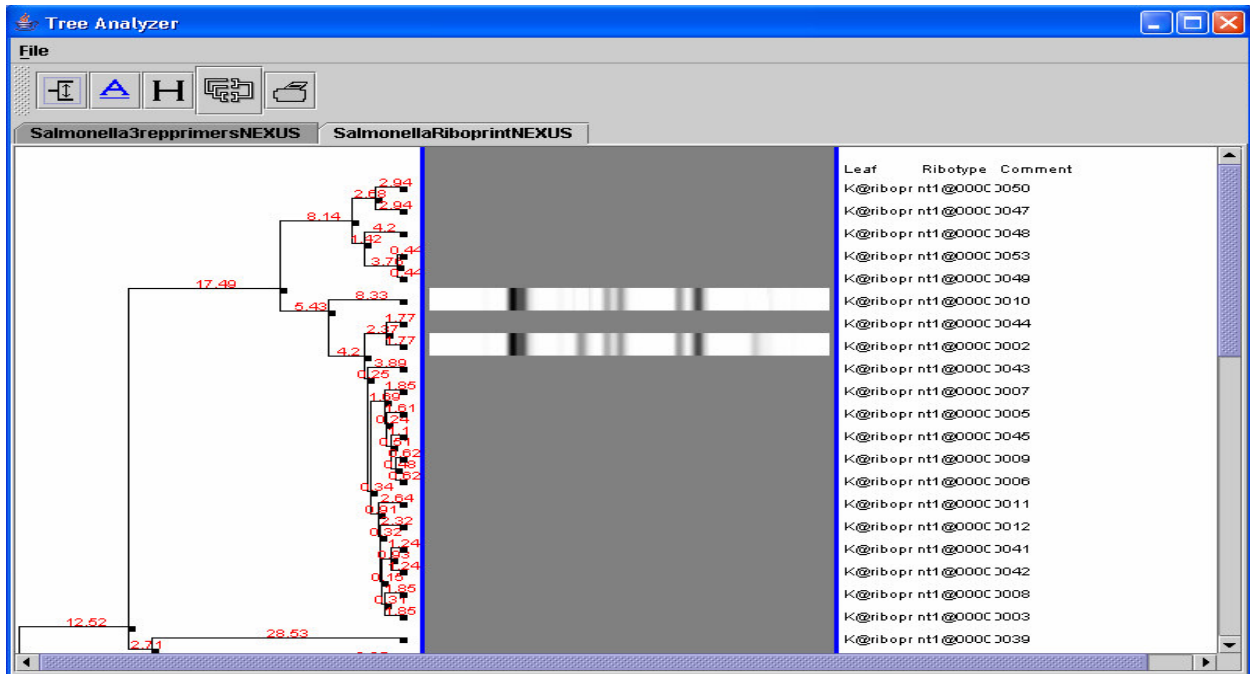
TreeAnalyzer is able to compare multiple trees and produce an exact solution. The similarity index, a numeric result, is also applied as a quantitative measurement of similarity and dissimilarity information for tree comparison.

However, there are still some issues that need to be addressed in the future. First, the branch length may be taken into consideration in computing MAST. Also, the visualization of MAST may be improved by using other techniques, such as 3D trees. With 3D tree-mapping techniques, the degree of similarity can be represented as the angle of the MASTs away from the plane, and multiple MASTs can be displayed at once. Finally, for some researchers, the similarity index does not provide enough information and some other phylogenetic tree comparison metrics such as the quartet metric or partition metric can be used as a complement to provide more information.

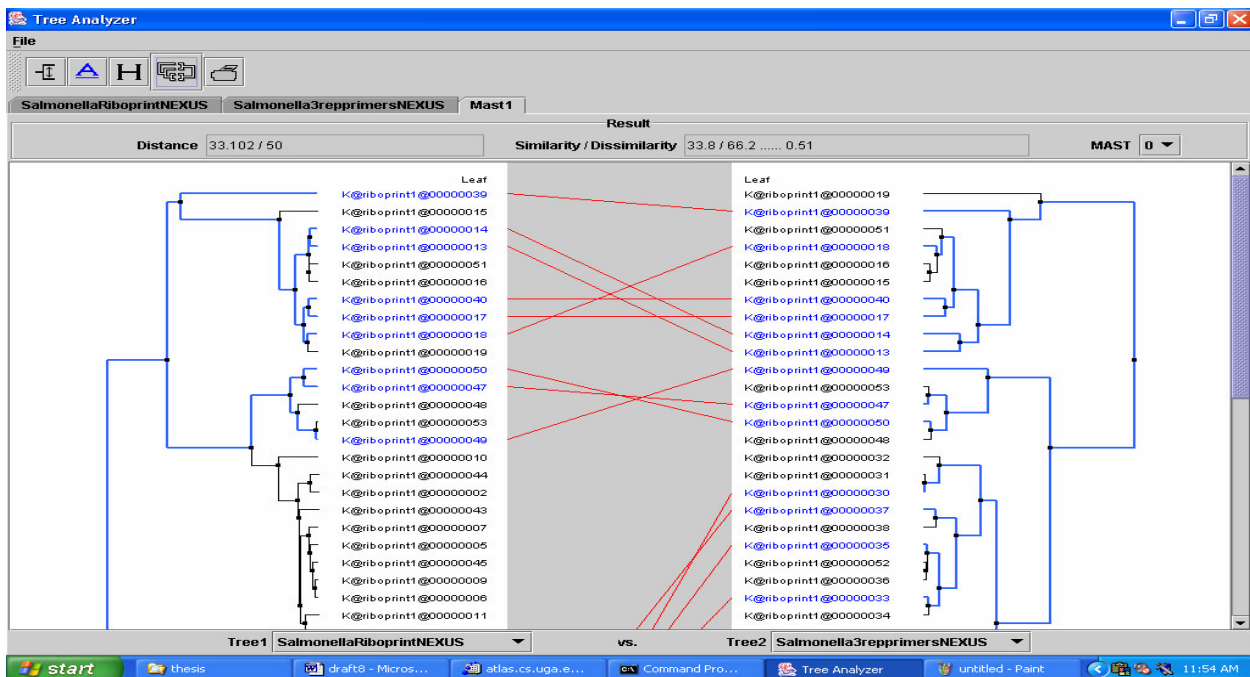
## 7. References

- [1] M. Farach, T. Przytycka and M. Thorup. The maximum agreement subtree problem for binary trees. *Proc. Of 2<sup>nd</sup> ESA*, 1995.
- [2] W. H. E. Day. Analysis of quartet dissimilarity measures between undirected phylogenetic trees. *Systematic Zoology*, 35:325-333, 1986.
- [3] M. Farach and M. Thorup. Optimal Evolutionary Tree Comparison by Sparse Dynamic Programming (Extended Abstract). *FOCS: 770-779*, 1994.
- [4] D. M. Hillis, C. Moritz, and B. K. Mable. *Molecular Systematics*. Sinauer Assoc. Inc, USA,ed, 1996.
- [5] D. Penny, M. D. Hendy and M. Steel. Progress with methods for constructing evolutionary trees. *Trends Ecol. Evol.* 7:73 – 79, 1992.
- [6] D. Penny and M. D. Hendy. The use of tree comparison metrics. *Syst. Zool.* 34: 75-82, 1985.
- [7] D. F. Robinson and L. R. Foulds. Comparison of phylogenetic trees. *Math. Biosci.* 53: 131-147, 1981.

- [8] G. F. Estabrook, F. R. McMorris and C. A. Meacham. Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units. *Syst. Zool.* 34:193-200, 1985.
- [9] M. S. Waterman and T. F. Smith. On the similarity of dendrograms. *J. theor. Biol.* 73: 789-800, 1978.
- [10] E. N. Adams. Consensus techniques and the comparison of taxonomic trees. *Systematic Zoology*, 21:390-397, 1972.
- [11] E. Kubicka, G. Kubicki and F. R. McMorris. An algorithm to find agreement subtrees, *Journal of Classification*, 12:91-99, 1995.
- [12] C. R. Finden and A. D. Gordon. Obtaining common pruned trees. *Journal of Classification*, 2: 255 – 276, 1985.
- [13] D. Bryant. Building trees, hunting for trees and comparing trees – Theory and methods in phylogenetic analysis. Ph. D. thesis, Department of Mathematics, University of Canterbury, NZ, 1997.
- [14] D. L. Swofford. When are phylogeny estimates from molecular and morphological data incongruent? Pages 295-333 in: *Phylogenetic analysis of DNA sequences (M. M. Miyamoto and J. Cracraft, eds.)*. Academic Press, New York, 1991.
- [15] D. R. Maddison, D. L. Swofford, and W. P. Maddison. NEXUS: An extensible file format for systematic information. *Syst. Biol.* 46:590-621, 1997.
- [16] Java 2 SDK, Standard Edition Version 1.4.1\_05. [http://java.sun.com/products/archive/j2se/1.4.1\\_07/README.html](http://java.sun.com/products/archive/j2se/1.4.1_07/README.html)
- [17] M. Farach, T. M. Przytycka and M. Thorup. On the Agreement of Many Trees. *Information Processing Letters*, 55:297-301, 1995.
- [18] W. Goddard, E. Kubicka, G. Kubicki and F. R. McMorris. The Agreement Metric for Labeled Binary Trees, *Mathematical Biosciences* 123: 215-226, 1994.
- [19] D. L. Swofford. PAUP\*. *Phylogenetic Analysis Using Parsimony (\*and Other Methods)*. Version 4. Sinauer Associates, Sunderland, Massachusetts, 2003.
- [20] J. Felsenstein. PHYLIP -- Phylogeny Inference Package (Version 3.2). *Cladistics* 5: 164-166, 1989.
- [21] M. Steel and T. Warnow. Kaikoura tree theorems: computing the maximum agreement subtree, *Information Processing Letters*, 48: 77-82, 1993.
- [22] M. Farach and M. Thorup. Fast comparison of evolutionary trees (extended abstract), in *Proc. 5<sup>th</sup> Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia*, pp. 481-488, 1994.
- [23] R. Cole, M. Farach, R. Hariharan, T. Przytycka and M. Thorup. An  $O(N \log N)$  algorithm for the maximum agreement subtree problem for binary trees, *SIAM J. Comput.* 30 (5): 1385-1404, 2000.
- [24] C. J. Lawrence, C. M. Zmasek, R.K. Dawe, and R.L. Malmberg. 2004. LumberJack: a heuristic tool for sequence alignment exploration and phylogenetic inference, *Bioinformatics*, in press.
- [25] M. Wilkinson. REDCON 3.0: software and documentation. Department of Zoology, The Natural History Museum, London.
- [26] F. J. Rohlf. 2002. NTSYSpc: Numerical Taxonomy System, ver. 2.1. Exeter Publishing, Ltd.: Setauket, NY
- [27] R. D. M. Page. 1993. *COMPONENT: Tree comparison software for Microsoft Windows, version 2.0*. The Natural History Museum, London.
- [28] W. P. Maddison and D. R. Maddison. 2003. Mesquite: a modular system for evolutionary analysis. Version 1.0 <http://mesquiteproject.org>
- [29] A. Amir and D. Keselman. Maximum Agreement Subtree in a Set of Evolutionary Trees: Metrics and Efficient Algorithms. *SIAM Journal of Computing*, Volume 26, Number 6, pp:1656 – 1669, 1997.



**Figure 7. Data input and display tree.** The tree name is displayed as the panel name. The tree topology with branch lengths is displayed on the left side; the corresponding sequence name, type and comments are shown on the right side; the images of some sequences are displayed in the center.



**Figure 8. Visualized MASTs.** Two trees from the compared group are selected to display the results; users may select other tree pairs. If multiple MASTs exist, the user may select other MASTs. Leaves and branches that belong to the selected MAST are highlighted. The leaves are also connected by red lines to show the common parts of the two trees. The tree distance (the number of leaves pruned to get MASTs) and the similarity index are computed as well.

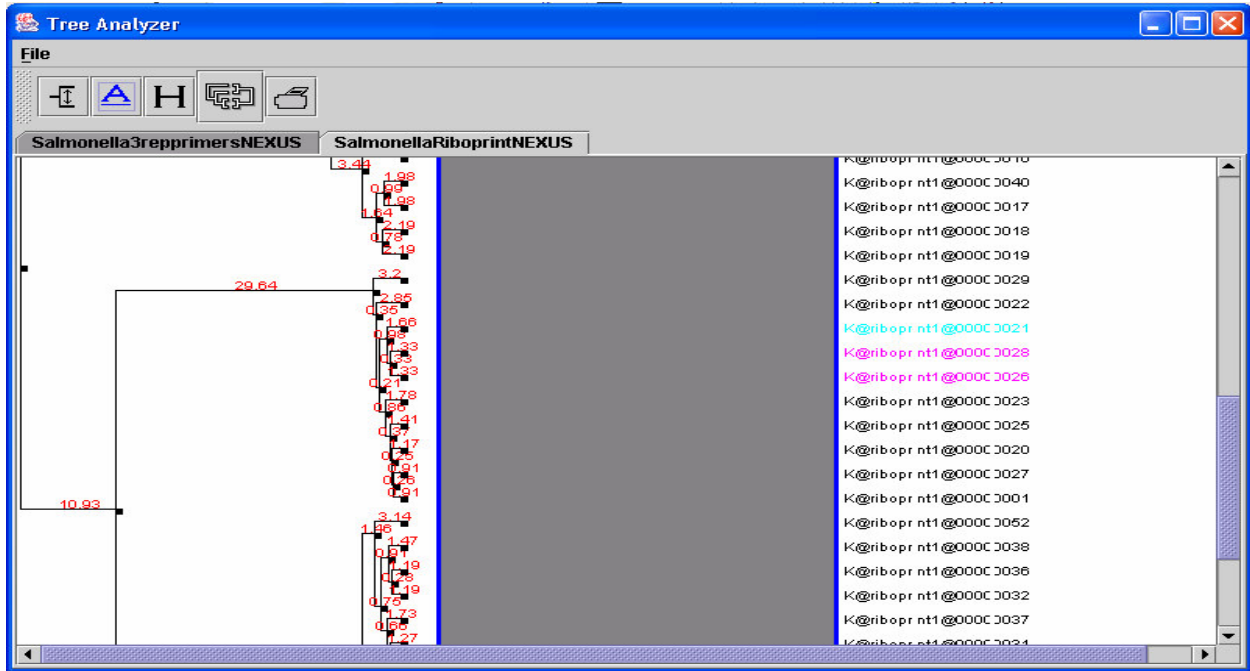


Figure 9. Tree structures of "Salmonella3reprimers" and "SalmonellaRiboprint" displayed by TreeAnalyzer.