Nodal Distance Algorithm: Calculating a Phylogenetic Tree Comparison Metric

John Bluis and Dong-Guk Shin
Computer Science and Engineering
University of Connecticut
Storrs, CT 06269-3155, USA
email: robert.bluis@uconn.edu shin@engr.uconn.edu

Abstract

Maintaining a phylogenetic relationship repository requires the development of tools that are useful for mining the data stored in the repository. One way to query a database of phylogenetic information would be to compare phylogenetic trees. Because the only existing tree comparison methods are computationally intensive, this is not a reasonable task. Presented here is the nodal distance algorithm which has significantly less computation time than the most widely used comparison method, the partition metric. When the metric is calculated for trees where one species has been repositioned to a distant part of the tree no further computation is required as is needed for the partition metric. The nodal distance algorithm provides a method for comparing large sets of phylogenetic trees in a reasonable amount of time.

KEYWORDS*: tree comparison, phylogenetic tree, algorithm*

Introduction

A major goal of molecular evolution studies is creating and evaluating phylogenetic trees. One of the most valuable uses of phylogenetic trees is that they may provide insight into the variation that occurs in a number of gene, protein, and genome characteristics. Paulsen, et al. [1] explain how the relationships between species can be used to predict gene functions. With a large number of completely sequenced genomes and many more in progress, there is large publicly available dataset that can be used for an automated process of building phylogenetic trees.

Building a repository of phylogenetic trees is not a simple task for a number of reasons. Meaningful relationships between species using sequence data is only valuable if the sequences are known orthologs or paralogs [2]. Hillis, et al [3] and Maley and Marshall [4] show how it is often difficult to produce an accurate phylogenetic tree. Even if a database of phylogenetic trees is made available its usefulness will be measured by how it can be queried. One valuable way of querying a database with phylogenetic data would be to compare the relationships that occur in the phylogenetic trees by using a tree comparison method

Before deciding to develop a repository of phylogenetic data, we discovered that the lack of a good comparison tool would severely limit the usefulness of this type of endeavor and questions if it is worthwhile to spend resources to store and upkeep this type of data. While researching the possibility of developing this type of system we found that there are no appropriate algorithms for calculating the differences that exist between phylogenetic trees in a reasonable amount of time. However, we were able to develop a new algorithm with a reasonable complexity. With this algorithm available, we felt justified in building a repository of phylogenetic trees and a software tool for browsing and querying the database [5]. Querying the database for similar relationships is valuable based on the assumption that if the variation between two data sets is identical, it may have been caused by the same factors or at least has been influenced by similar factors. Being able to locate similar variation and then create an open-ended search tool for discovering useful resources are the two major goals of our endeavor.

Comparison Metrics

In order to produce a metric that accurately measures the difference between phylogenetic trees we need to answer the question, what does it mean for two trees to be similar? Conceptually, we would say that two trees are similar if they convey the relationships between a set of species in a similar way (taxonomic congruence). This conceptual definition can be translated into a quantitative measure in a number of different ways.

There have been a number of proposed methods for making tree comparisons [6-13]). The most commonly used metric is the partition metric introduced by Penny and Hendy [9] and the metric from the crossover method established by Robinson [6]. The partition metric is similar to the crossover method that was later generalized by Robinson and Foulds [7-8]. The partition metric

Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering (BIBE'03)
0-7695-1907-5/03 $17.00 © 2003 IEEE

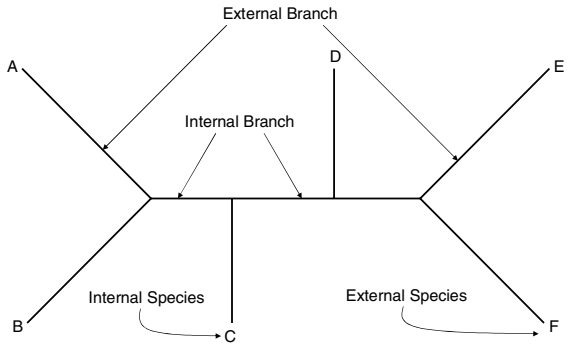


Figure 1. Tree terminology for a phylogenetic tree.

compares the partitions, or splits, that occur in two different trees. A partition is created when an internal branch of a tree (a branch with no leaf nodes, see terminology in Figure 1) is removed from the tree. This results in two sub-trees each having a unique subset of species (Figure 2). Penny and Hendy explain a simple way to calculate the partition metric by associating a binary number (1, 2, 4, 8, etc.) to each taxon. By summing the value of all the species in a subset, each unique subset will have a unique sum of the binary numbers associated with its members. Since each partition creates two unique subsets, only one subset needs to be associated with any partition. The smaller sum becomes the value associated with that partition. This is a computationally expensive way of characterizing each tree, but using the defining splits of a tree is useful for uniquely identifying individual trees. Our notation throughout this paper uses the node description

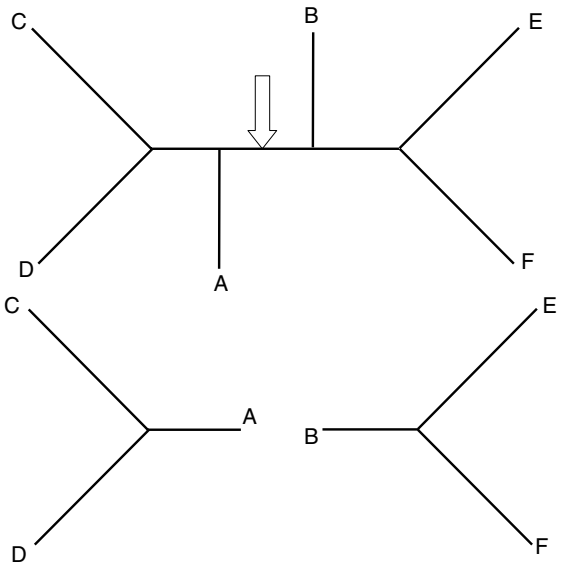


Figure 2. Creating sub-trees for the partition metric.

information to define each subset. The subset with fewer members, or the subset that comes first alphabetically when there are an equal number of species in each split, is used as the defining split. Therefore, the partitions from the tree in figure 2 are {CD}, {EF}, and {ACD}. So this tree can be represented by the notation {CD}:{EF}:{ACD}.

To compare two trees all the partition values from each tree are determined and then compared. The partition metric is defined as the number of subsets from one tree that do not match a subset from the other tree. According to the partition metric trees 1 and 2 from Figure 3 are very similar in that they only differ in one of their partitions: AB and EF are identical and they differ in the split ABC and ABD. So the partition metric equals 1 when comparing trees 1 and 2. By calculating the partitions in tree 3 and tree 4 we see that no partitions in tree 1 are similar to them and so they have the maximum partition metric possible, 3. However, the only difference that tree 3 has with tree 1 is that species B has been moved to the opposite end of the tree. So it seems as if there are some similarities between the two trees yet this would not be revealed from using this metric. On the other hand, there seems to be very little, if any, similarity in the

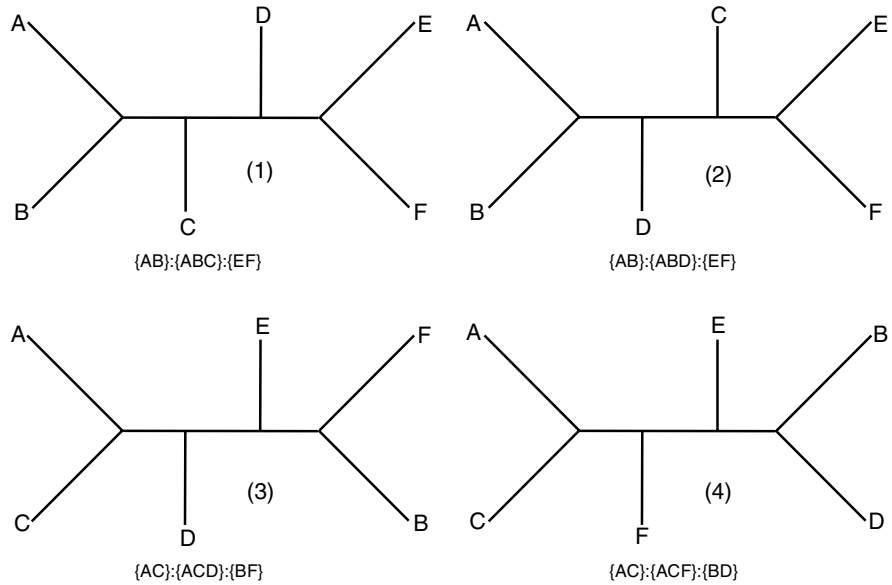


Figure 3. Sample phylogenetic trees with partition notations.

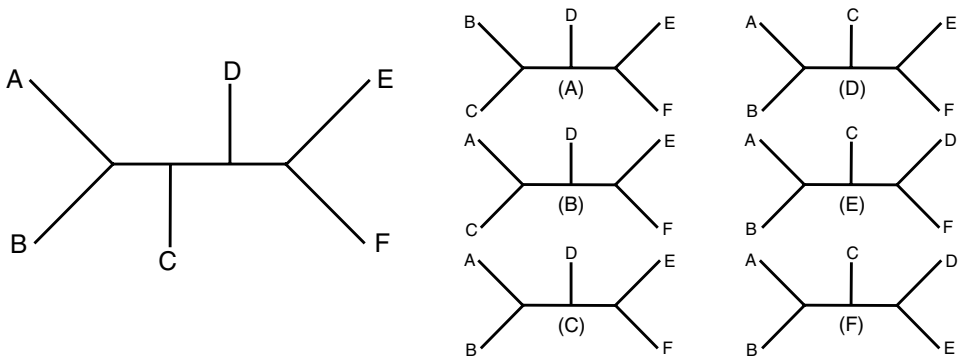


Figure 4. Removing species from tree 1.

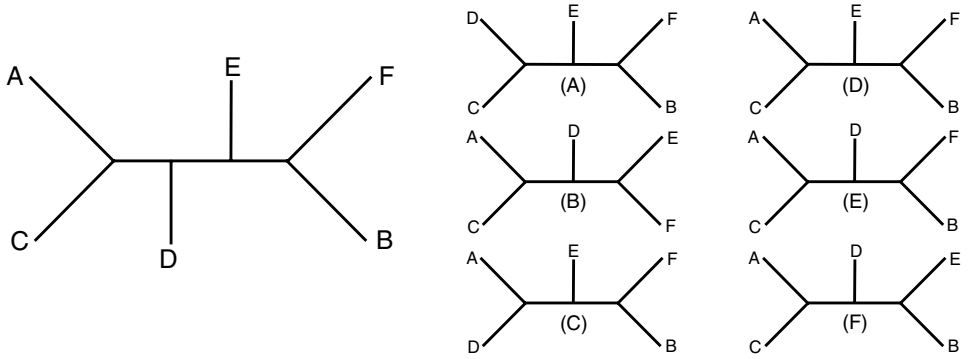


Figure 5. Removing species from tree 3.

topology of trees 1 and 4. Because of this it is illogical to place trees 3 and 4 in the same category of dissimilarity. Tree 3 reflects more similarity in the relative placement of its species when comparing it to tree 1 then it does when comparing it to tree 4.

To help alleviate the effect this has on the partition metric, the authors present a method for determining a more accurate metric for two trees that may have some similarities. By removing species from the trees that are being compared we can determine if there are just a small number of deviant species (Figure 4 and Figure 5). For example, if species B is removed from tree 1 and tree 3 the remaining tree topology is identical. We see that tree B in Figure 4 and 5 are identical. However, in order to determine the best species to remove from the tree, a new partition metric must be re-calculated by trying this method for every species in the tree as has been done for tree 1 in Figure 4 and tree 3 in Figure 5. If no significant improvement is found, then this step must be repeated for each of the sub-trees A through F that was created. Although this method does work to an extent, it is a nightmare computationally. This could lead to n! partition metrics being calculated for each two trees that are being compared. If it was possible to determine what trees that this analysis needs to be done for, it may be acceptable for use over a large set of trees, but we do not have this type of a priori knowledge.

Another difficulty with this method is that the metric is not normalized. By reducing the number of species in the tree, the maximum partition metric is automatically reduced by 1. The maximum metric between trees with 6 species is 3, but for 5 species the maximum is 2. So the question that is raised is, does it mean the same thing when we get the same value after removing more species in one tree than from another tree? Because the size of the tree is a factor in the value of the partition metric the meaning of the new metric value is unclear. One final difficulty would be in attempting to define an appropriate place to stop removing species. This would require describing when a "significantly" small metric has been discovered.

Nodal Distance Algorithm

We have developed an algorithm for comparing phylogenetic trees that is better suited for analyzing large sets of data. As with most comparisons, we assume that the two trees have the same exact set of species. The algorithm is as follows:

Algorithm Nodal Distance Metric: Find the increase in the differences of distance between species in two phylogenetic trees.

Input: Phylogenetic trees, P_1 and P_2, each of which has the same set of species.

Output: Sum of all distance differences between P_1 and P_2.

Method:

//Phase 1: Compute two nodal distance //matrices, $Distance_{ij}(P_k)$ where k = 1, 2. Let //S = (s_1, s_2, …, s_n) be the ordered set of all n //species found in the input trees. Initially //$Distance_{ij}(P_k)$ is set to 0 for all i, j, and k

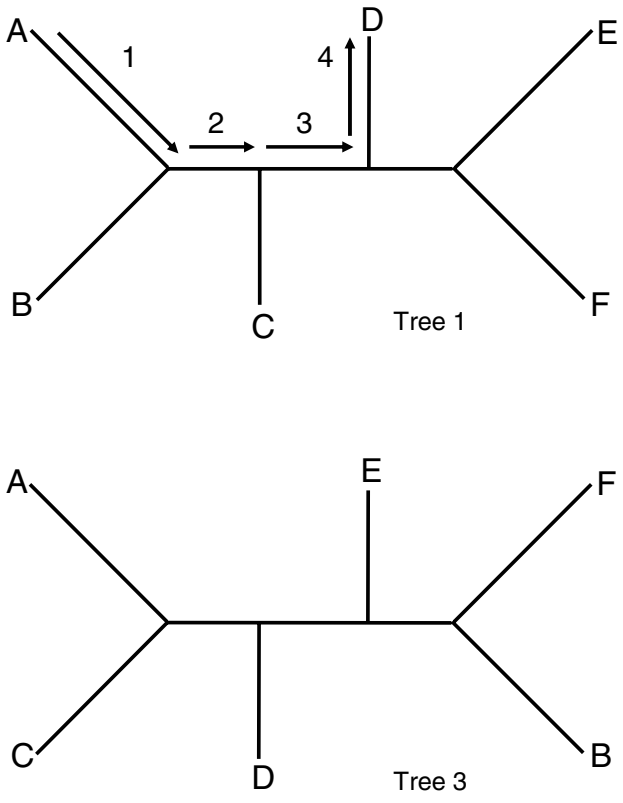
```
//combinations where i and j range from 1 to
//n and k = 1, 2.

BEGIN
for P_k where k=1 and k=2 do the following:
(1)   for (i = 1; i < n; i++) {
(2)     node_i = node with value s_i;
(3)     level_i = the level in the tree that node_i
is in
(4)     for (j = i + 1;  j <= n; j++) {
(5)       node_j = node with value s_j;
(6)       level_j = the level in the tree that node_j
is in
(7)       while (level_i != level_j) {
(8)         if (level_i < level_j) then level_j++;
(9)         else level_i++;
(10)        Distance_ij++;
(11)      }
(12)      while (node_i != node_j) {
(13)        node_i = parent_i;
(14)        node_j = parent_j;
(15)        Distance_ij = Distance_ij + 2;
(16)      }
(17)      if (level_i == 0) then distance_ij--;
(18)    }
(19)  }

// Phase 2: Find the difference between P_1
//and P_2 using Distance_ij which contains the
//nodal distances between all species i and j
i//n each tree.

(20)  for (i = 1; i < n; i++) {
(21)    for (j = 1; j <= n; j++) {
(22)      nd-metric := nd-metric +
|Distanceij(P_1) – Distanceij(P_2)|;
(23)    }
(24)  }
END
```

The algorithm analyzes the change that occurs in the relative positions of the species in a tree. The only input is two phylogenetic trees, P_1 and P_2, which are to be compared. To measure the change in the relative position of each of the species in P_1 and P_2 we first count the branches along the path from one species node to another species node in the tree. This is the same as the number of new nodes that are reached in this same path. This value we refer to as the nodal distance, which is computed in steps 1-19 of the algorithm. This value needs to be calculated for every possible combination of two species in each of the two trees. Our implementation of the algorithm converts a text-based representation of a tree (in Newick format) into an implicit representation of a tree structure. By providing the parent node number and the species associated with that node of the tree, if one exists, we were able to efficiently calculate the nodal distances.



Species	Tree 1 Distance	Tree 3 Distance	Difference
AB	2	5	3
AC	3	2	1
AD	**4**	3	1
AE	5	4	1
AF	5	5	0
BC	3	5	2
BD	4	4	0
BE	5	3	2
BF	5	2	3
CD	3	3	0
CE	4	4	0
CF	4	5	1
DE	3	3	0
DF	3	4	1
EF	2	3	1

Figure 6. Calculating nodal distance.

The nodal distance between species i and j in the tree is calculated by following a $path_i$ and a $path_j$ up the tree via parent nodes until the paths reach a common node. The value is stored in the array matrix $Distance_{ij}(P_1)$ for tree P_1 and in the array matrix $Distance_{ij}(P_2)$ for tree P_2. Each time $path_i$ or $path_j$ is extended to include another node the nodal distance is increased by one. While traversing the tree, each path will be at some node that will be denoted as $node_i$ for $path_i$ or $node_j$ for $path_j$. Once the nodes labeled with species i and j are found, if the level of one node is lower than level of the other node, a path up the tree must be taken until the level of $node_i$ equals the level of $node_j$. In this case, the nodal distance will increase by one for each level that the path moves to. Once both nodes are at the same level, we can then continue up the tree via parent nodes until each path leads to a common node. Each time both paths move up to the next level, the distance is increased by two. Each path is extended up the tree until the two paths reach a common node. Once this condition has been met phase one is complete. Before moving on to phase 2, we must check for an exception that occurs if the paths end at the root node. Because the root node is not a true node in the unrooted phylogenetic trees P_1 and P_2, we must decrease the distance by one. The root node links the two halves of the tree together. Moving one of the paths up to the root node connects the paths of the tree.

At this point we have a set of nodal distances stored in the arrays $Distance_{ij}(P_1)$ and $Distance_{ij}(P_2)$. This is required for continuing on to phase 2 (steps 20-24) of the algorithm. We can calculate the differences in the nodal distances between each pair of species since each tree contains the same set of species. By summing all of the differences, we calculate a value that reflects the amount of change that has occurred in the relative positions of the species in the tree. We call this value the nodal distance metric or nd-metric. The nd-metric represents the relative changes in species position or topology that one tree has relative to another tree. This is because the metric measures how much farther or closer each species is to every other species in the trees.

Figure 6 shows how the nodal distance is calculated in a sample tree. The nodal distance is the number of branches that are part of the path from the node where A is to the node where D is. The arrows in the tree enumerate the path between the species A and D showing that the nodal distance between A and D is equal to 4. The tables seen in Figure 6 contain the calculated values of all nodal distances in the two trees.

Once the nodal distances of two trees have been calculated, the distances are used to calculate the nd-metric. Because tree comparisons are done with trees containing identical sets of species, there will be identical

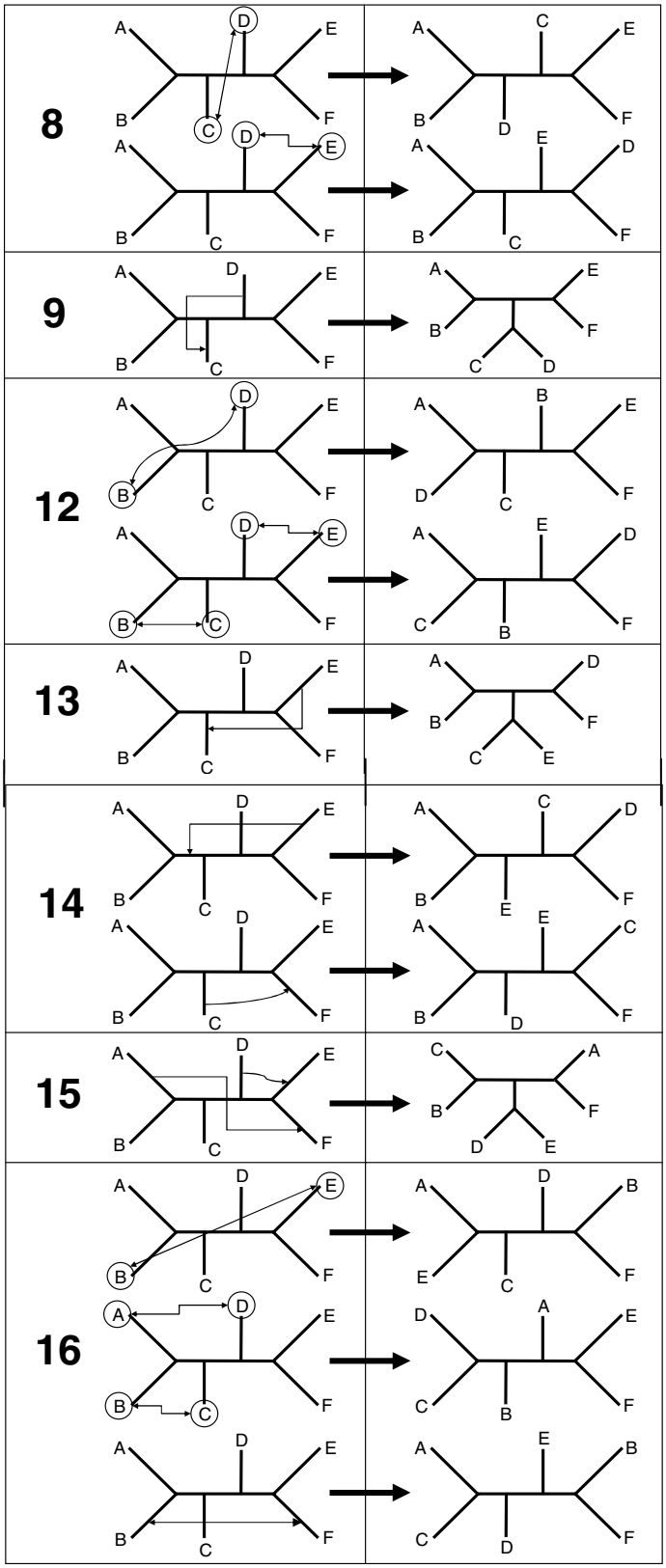


Figure 7. Visual representation of classification schemes from table 1.

Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering (BIBE'03)
0-7695-1907-5/03 $17.00 © 2003 IEEE

Tree Topology	nd-metric	Classification (based on change to template tree)
AB:DC:EF AB:CE:DF AB:CF:DE AC:BD:EF BC:AD:EF	8	Species with a nodal distance of 3 swapped (A-C, B-C, C-D, D-E, D-F)
AB:[CD]:EF	9	Internal branches fused (C to D or vice versa) & change in topology of tree
AB:ED:CF AB:FD:CE AC:BE:DF AC:BF:DE AD:CB:EF BC:AE:DF BC:AF:DE BD:CA:EF	12	Species with a nodal distance of 4 swapped (C-E, C-F, B-D, A-D) OR 2 sets of species with a nodal distance of 3 swapped (B-C & D-E, B-C & D-F, A-C & D-E, A-C & D-F)
AB:[CE]:DF AB:[CF]:DE AC:[BD]:EF AD:[BC]:EF	13	External species moved to branch of species that have a nodal distance of 4 (A to D, B to D, E to C, F to C) & change in topology of tree
AB:EC:DF AB:FC:DE AB:DE:CF AB:DF:CE AC:DB:EF AD:BC:EF BC:DA:EF BD:AC:EF	14	External species moved to the furthest internal branch (E to AB, F to AB, A to EF, B to EF) OR Internal species moved to an external branch (C to F, C to E, D to A, D to B)
AC:[BE]:DF AC:[BF]:DE AF:[BC]:DE	15	External branch to opposite external branches & new internal branches fused
AB:EF:CD AB:FE:CD AC:DE:BF AC:DF:BE AE:BC:DF AE:CD:BF AE:DC:BF AE:FD:BC AF:BC:DE AF:CD:BE AF:DC:BE AF:ED:BC BE:AC:DF BF:AC:DE CD:AB:EF CD:BA:EF	16	One set of external species swapped with internal branches (AB internal in any order & CD external or EF internal in any order and CD external) OR External species moved to branch of opposite set of external species (A to E, A to F, B to E, B to F, E to A, E to B) OR Swap external species (A-E, A-F, B-E, B-F)
AE:[BF]:CD AF:[BE]:CD	17	Internal branches fused & external species (in original tree) swapped
AC:ED:BF AC:FD:BE AD:CE:BF AD:CF:BE AE:CB:DF AE:BD:CF AE:FC:BD AE:DF:BC AF:CB:DE AF:BD:CE AF:EC:BD AF:DE:BC BE:CA:DF BE:AD:CF BF:CA:DE BF:AD:CE	18	Internal species swapped & one external species moved to branch of opposite set of external species OR External species swapped with adjacent internal branch & external species moved to branch of opposite set of external species
AD:[BE]:CF AD:[BF]:CE AE:[BD]:CF AF:[BD]:CE	19	External branch to farthest internal branch & new external branches swapped
AC:EB:DF AC:FB:DE AC:EF:BD AC:FE:BD AD:BE:CF AD:EB:CF AD:BF:CE AD:FB:CE AD:EC:BF AD:FC:BE AD:EF:BC AD:FE:BC AE:DB:CF AE:BF:CD AE:FB:CD AE:CF:BD AF:DB:CD AF:BE:CD AF:EB:CD AF:CE:BD BC:FA:DE BD:AE:CE BD:EA:CF BD:AF:CE BD:FA:CE BE:DA:CF BE:AF:CD BE:FA:CD BF:DA:CE BF:AE:CD BF:EA:CD CE:AB:DF CE:BA:DF CF:AB:DE CF:BA:DE	20	A number of different combinations where four or five of the species have been removed & reinserted at new locations in the tree.
AE:[BC]:DF	21	External branch to opposite external branches & new internal branches fused

Table 1. Classification of the 105 possible 6 species phylogentic trees.

sets of nodal distances from each of the trees. In the last column of the table in Figure 6, we show the values of the difference in nodal distance for each set of species. The sum of the values in this column is the nd-metric and is equal to 16 for these two trees.

Discussion

There are a number of advantages that the nodal distance algorithm presents over the other methods. One major advantage is that the algorithm we have presented only requires the metric to be calculated once for any tree. Because the partition metric requires re-calculation of the metric for each tree, at minimum it would take several weeks to return results of a query on a database with 100,000 phylogenetic trees. We have been able to run queries on over 5,000 trees that contain 21 or less species in less than two minutes.

A second advantage is that no species need to be removed from the trees so metrics for trees of the same size are normalized. No species need to be removed from the trees because the nd-metric evaluates the relative change in species positions in the tree. For example, using the same trees from Figure 3 that we used to show how the partition metric is calculated, we can see that no other changes need to be made to the nd-metric to show that tree 3 has more similarities to tree 1 than tree 4 does. The nd-metric equals 8 for trees 1 and 2, 16 for trees 1 and 3, and 20 for trees 1 and 4. The nd-metric indicates tree 1 is more similar to tree 3 than it is to tree 4.

Although tree 3 has a lower nd-metric than tree 4, with respect to tree 1, we must be sure that this accurately reflects the amount of similarity between trees in general. For example, for what other trees does the nd-metric evaluate to 16? We show in Table 1 that we can classify the 105 different 6 species unrooted phylogenetic trees based on their nd-metric value. To answer the question we have posed, the trees with a nd-metric value that equals 16 have unique characteristics and these characteristics are not found in any trees that evaluate to a different nd-metric value. We are able to classify trees based on what changes must occur in tree 1 to create the tree it is being compared with. The change that occurs within tree 1 to transform it into a tree with an nd-metric of 16 is one of the following: i) one set of external branches (meaning they are next to each other) are exchanged with the two internal branches, or ii) one external species is moved to the branch of an external species on the opposite end of the tree (tree 3), or iii) external species from opposite ends of the tree are swapped.

We have shown in Figure 7 sets of trees with a number associated with each set. The numbers are the nd-metric values that are calculated when tree 1 is compared with any of the trees in the group. We have included all possibilities for trees evaluating to a nd-metric less than or equal to 16. For example, Converting tree 1 to tree 3 is the same as the change that is occurring in the bottom tree from group 16. This tree is showing that an external species is being removed and reinserted on an external branch at the opposite end of the tree. This means that B is removed and placed on the branch with species E or species F, B is removed and placed on the branch with species E or species F, E is moved to A or B, or F is moved to A or B. Table 1 classifies all 105 possible six species trees based on the terminology of Figure 1. We have not fully described the trees with nd-metric values greater than 16 since they are typically uninteresting and the trees are essentially dissimilar.

An advantage of the nd-metric over the partition metric is that it is robust in its ability to separate the trees into more distinct classes. Trees with a partition metric equal to 2, have an nd-metric value of 8 or 9. Trees with a partition metric equal to 1 have nd-metric values equal to 12, 13, 14, or 16. Only when the nd-metric results in a value of 16 are there mixed values of the partition metric. The two trees that have a partition metric of 1 and an nd-metric of 16 require more changes than any of the other trees with a partition metric of 1 to get a topology identical to tree 1.

The main benefit of classification based on the nd-metric is the added feature it provides for users of our program. When querying the database of phylogenetic trees we would like to provide the user with the option of setting a stringency level for their search. If the user only desires the program to return trees that are significantly similar to the input tree, he or she can use the nd-metric classifications to determine the type of trees that would be returned to them in the results. This is in contrast to many tools that require a trial-and-error system to determine what a good value for a threshold would be. If the user only wants to know the trees that show a specific change

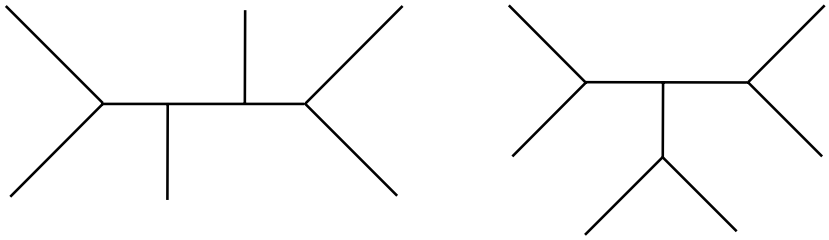


Figure 8. Possible tree topologies for 6 species trees.

in the topology, this type of query could be presented, as well.

We would like to determine if there is a way of classifying trees with n species based on the value of the nd-metric. One problem that may be difficult to work around in classifying the trees is that there are different basic tree topologies for trees with a given number of species (Figure 8). The classification scheme we are using may not work since with larger number of species, and hence a larger number of different topologies, the classifications may be too specific. We may need a more general format of classifying homogeneous groups or the number of different possible classes may become too large. However, the nd-metric's capability of determining the differences in the amount of change within a tree would not be affected if this could not be accomplished.

Summary

We have described a new method for comparing phylogenetic trees. The nd-metric, which measures the difference in relative positions of species, is normalized and handles some of the difficulties that have previously been encountered when developing tree comparison metrics. There is a possibility that the metric can be utilized for classifying different topologies of phylogenetic trees, however a more in-depth study of the mathematical properties that these classifications are based on is needed. The nodal distance algorithm is utilized in our phylogenetic repository system and can quickly return queries over the relationships between species expressed in the stored phylogenetic trees.

References

[1] Paulsen, I.T., *et al.* (1998) Microbial genome analyses: global comparisons of transport capabilities based on phylogenies, bioenergetics and substrate specificities. *J. Mol. Biol.* 277:573-592.

[2] Tatusov, R. L., *et al*. (1997) A Genomic Perspective on Protein Families. *Science* 278:631-637.

[3] Hillis, D. M. *et al*. (1994) Application and Accuracy of Molecular Phylogenies. *Science* 264:671-677.

[4] Maley, L. E. and Marshall, C.R. (1998) The Coming of Age of Molecular Systematics. *Science* 279:505-506.

[5] Bluis, J., *et al.* (2001) Comparing trees in a phylogenetic relationship repository. *Second IEEE International Symposium on Bioinformatics and Bioengineering.* pp 166-173.

[6] Robinson, D.F. (1971) Comparison of labeled trees with valency three. *J. Comb. Theory* 11:105-119.

[7] Robinson, D.F. and Foulds, L.R. (1979) Comparison of weghted labelled trees. Pp. 119-126 in *Lecture Notes in Mathematics*, Vol. 748. Springer-Verlag, Berlin.

[8] Robinson, D.F. and Foulds, L.R. (1981) Comparison of phylogenetic trees. *Math. Biosci.* 53:131-147.

[9] Penny, D. and Hendy M.D. (1985) The use of tree comparison metrics. *Syst. Zool.* 34:75-82.

[10] Waterman, M.S., and Smith, T.F. (1978) On the similarity of denrograms. *J. Theor. Biol.* 73:789-800.

[11] Farris, J.S. (1973) On comparing the shapes of taxonomic trees. *Syst. Zool.* 22:50-54.

[12] Estabrook, G.F., *et al.* (1985) Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units. *Syst. Zool.* 34:193-200.

[13] Crozier, R.H., *et al.* (1986) Evolutionary patterns in some putative Australian species in the ant genus *Rhytidoponera. Aust. J. Zool.* 34:535-560.