Tree Distance

Raghuveer Gummadi
Computer Science
Georgia State University
Atlanta, Georgia
rgummadi1@student.gsu.edu

Eunice Olorunshola
Computer Science
Georgia State University
Atlanta, Georgia
eolorunshola1@student.gsu.edu

Vijaya Lakshmi Kocherla
Computer Science
Georgia State University
Atlanta, Georgia
vkocherla1@student.gsu.edu

Sahid Kebe
Computer Science
Georgia State University
Atlanta, Georgia
skebe1@student.gsu.edu

***Abstract*— Billera Holmes Vogtmann Space is a metric space that allows researchers to quantitatively evaluate distance between trees. We implement an algorithm to measure geodesic distance between trees and present the results.**

***Keywords*— Phylogenetic Trees, BHV, Geodesic Distance , Topology, Newick, Metric space, Partition , GTP.**

I. AREA

A. Motivation

Phylogenetic trees describe the evolutionary history of organisms, and researchers construct these trees using a variety of different methods. This can result in several possible trees for the same set of organisms. For example, parsimony, maximum likelihood, distance based. Thus we need a way to quantitatively compare different phylogenetic trees. One method to this is through BHV (Billera, Holmes, Vogtmann) Space, and its related distance measure, geodesic distance, which describes the differences between the trees. Another such method is Wald space, which is outside the scope of this paper but is also an interesting metric space for trees.[3]

B. Challenge

BHV is a metric space that uses tree topology and edge lengths to plot a given tree's position in the space[1], and geodesic distance denotes the shortest possible distance between these trees in this metric space[1]. In this paper, we will explain how this tree space is constructed and demonstrate the use of Owen and Provan's algorithm to determine geodesic distance in polynomial time[2]. Thus the challenge of this paper will be to show the results of running this algorithm and to help the reader understand how it relates to the physical structure of the tree.

II. MODEL

1. TREE SPACE

A phylogenetic tree is a specific type of mathematical graph, which describes the evolutionary history of a set of organisms, with the leaf vertices representing the organisms and the interior vertices representing points at which the evolutionary history branches. To define, a *phylogenetic n-tree* is a tree T = (X, Ɛ, Σ), where X={0,1,2,...,n} is a labeled set of vertices, called leaves, of degree 1, and Ɛ is the set of interior (non leaf) edges, such that each interior vertex of T has degree at least 3, and Σ is the set of splits of the set X induced by the interior edges. In other words, the split associated with an edge e ∈ E represents the partition of X introduced by removing the edge e from T.

Inorder to find the distance between any two phylogenetic trees, we consider a phylogenetic tree space which is a collection of all possible trees for a set of species. This tree space is called BHV tree space which is introduced by Billera, Holmes, and Vogtmann.[1]

BHV space is a CAT(0) continuous space that represents trees with edge weights with an intrinsic *geodesic distance* measure. The BHV space of weighted trees can be embedded in a Euclidean space, where each distinct tree topology is associated with a Euclidean region, called an *orthant.* Trees with the same topology but different edge lengths belong to the same orthant and are represented by distinct points within the same orthant. Whereas, trees with different topologies belong to different orthants. Within each orthant, the coordinates of a point correspond to edge lengths of the corresponding tree topology. This tree space has unique shortest paths between the points, called *geodesics*. The length of this shortest path gives us the *geodesic distance* between the trees[1].

In general, a metric space (X, d) is a set X with a distance function d which is called a *metric*, defined on X.

The distance function should satisfy the following conditions for all x, y ∈ X:

(1) d(x, y) ≥ 0 and d(x, y) = 0 if and only if x = y;

(2) d(x, y) = d(y, x) (symmetry);

(3) d(x, y) ≤ d(x, z) + d(z, x) (triangle inequality)

A metric space X is called a *geodesic space* if every pair of points x, y ∈ X can be joined by a continuous path of length d(x, y), and the image of such a path is called a *geodesic segment.*

2. GEODESIC DISTANCE

Geodesic distance is the unique shortest path distance between two points on a surface. Our surface in this case is the BHV space in which our trees reside[1].

III. PROBLEM FORMULATION

- **Given :** A set of trees
- **Find :** The geodesic distance between those trees.

IV. ALGORITHM

GEODESIC DISTANCE

Owen and Provan introduced a polynomial time algorithm for finding a geodesic path. Once the path is known, computing its length is a simple matter.[2]

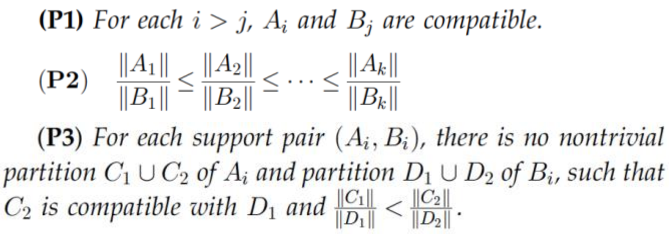
Since each orthant is locally a Euclidean space, the shortest path between two points within a single orthant is a straight line.

The difficulty comes in establishing which sequence of orthants joining the two topologies will contain the geodesic. In the case of four leaves, we could do this through a brute-force search, but we cannot hope to do so with larger trees[2].

Fig.1. shows the GTP algorithm for finding the geodesic path between two trees.

Following are the iterative steps for finding a geodesic between two trees T and T′:

1. Begin with some proper (T,T′)-path $\mathcal{T}^0$ with support $(\mathcal{A}^0,\mathcal{B}^0)$.
2. At each stage, we have a proper path $\mathcal{T}^\ell$ having support $(\mathcal{A}^\ell,\mathcal{B}^\ell)$ satisfying conditions (P1) and (P2). Check to see if $(\mathcal{A}^\ell,\mathcal{B}^\ell)$ also satisfies the condition (P3), and if not, create a new proper path $\mathcal{T}^{\ell+1}$ with support $(\mathcal{A}^{\ell+1},\mathcal{B}^{\ell+1})$ and having a smaller length than $\mathcal{T}^\ell$.
3. Continue until the geodesic is found.



(P1) *For each $i > j$, A_i and B_j are compatible.*

(P2) $$\frac{\|A_1\|}{\|B_1\|} \le \frac{\|A_2\|}{\|B_2\|} \le \cdots \le \frac{\|A_k\|}{\|B_k\|}$$

(P3) *For each support pair (A_i, B_i), there is no nontrivial partition $C_1 \cup C_2$ of A_i and partition $D_1 \cup D_2$ of B_i, such that C_2 is compatible with D_1 and $\frac{\|C_1\|}{\|D_1\|} < \frac{\|C_2\|}{\|D_2\|}$.*

EXPLANATION (1)

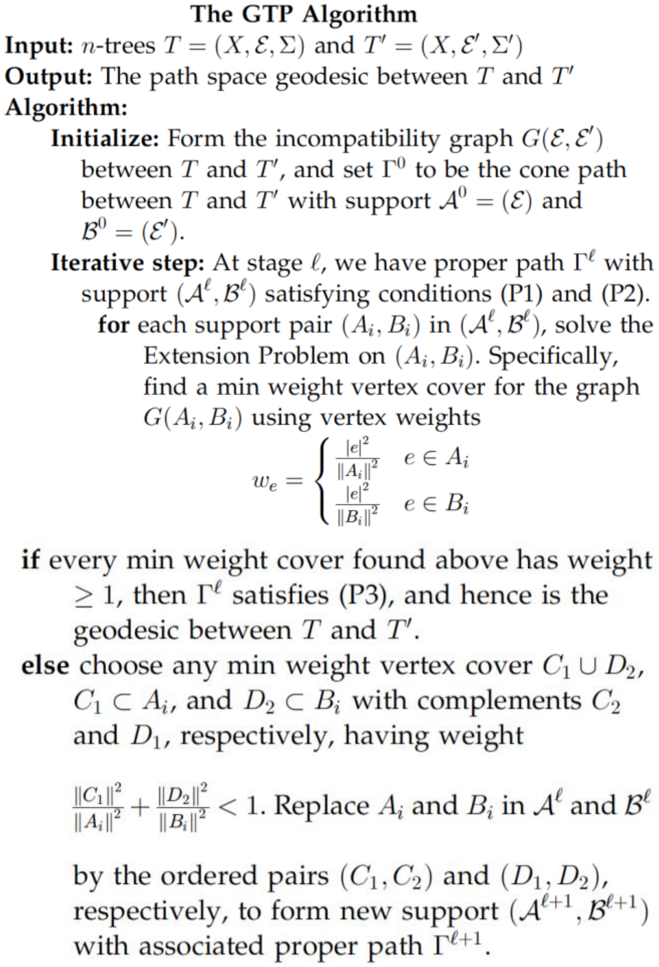
Property (P1) : means that the edge sets can only be compatible if every pair of the splits associated with A in Σ and B in Σ′ . Each of the edge sets in A and B have to form a union

A = $(A_1, \ldots \ldots, A_k)$ B = $(B_1, \ldots \ldots, B_k)$

as

Property (P2) : is the property that satisfies the conditions of a proper path space by having two n - trees , T = (X, ε, Σ) and T ′ = (X, ε' , Σ ′). The pair (A, B) are shown as two adjacent support pairs in a proper path space having their ratios equal.

Property (P3) : Non trivial partition means that the partition uses more than one subgroup and the subgroups are proper. This property simply means that for each support pair in (A_i, B_i) there is no partition that belong in $C_1 \cup C_2$of A_i and $D_1 \cup D_2$ of B_i . Therefore this property does not satisfy that C_2 and D_2 are compatible and the property (P1) .



The GTP Algorithm

Input: n-trees $T = (X, \mathcal{E}, \Sigma)$ and $T' = (X, \mathcal{E}', \Sigma')$

Output: The path space geodesic between T and T'

Algorithm:

```
    Initialize: Form the incompatibility graph $G(\mathcal{E}, \mathcal{E}')$
        between $T$ and $T'$, and set $\Gamma^0$ to be the cone path
        between $T$ and $T'$ with support $\mathcal{A}^0 = (\mathcal{E})$ and
        $\mathcal{B}^0 = (\mathcal{E}')$.
    Iterative step: At stage $\ell$, we have proper path $\Gamma^\ell$ with
        support $(\mathcal{A}^\ell, \mathcal{B}^\ell)$ satisfying conditions (P1) and (P2).
        for each support pair $(A_i, B_i)$ in $(\mathcal{A}^\ell, \mathcal{B}^\ell)$, solve the
            Extension Problem on $(A_i, B_i)$. Specifically,
            find a min weight vertex cover for the graph
            $G(A_i, B_i)$ using vertex weights
                $w_e = \begin{cases} \frac{|e|^2}{\|A_i\|^2} & e \in A_i \\ \frac{|e|^2}{\|B_i\|^2} & e \in B_i \end{cases}$
    if every min weight cover found above has weight
        $\ge 1$, then $\Gamma^\ell$ satisfies (P3), and hence is the
        geodesic between $T$ and $T'$.
    else choose any min weight vertex cover $C_1 \cup D_2$,
        $C_1 \subset A_i$, and $D_2 \subset B_i$ with complements $C_2$
        and $D_1$, respectively, having weight
        $\frac{\|C_1\|^2}{\|A_i\|^2} + \frac{\|D_2\|^2}{\|B_i\|^2} < 1$. Replace $A_i$ and $B_i$ in $\mathcal{A}^\ell$ and $\mathcal{B}^\ell$
        by the ordered pairs $(C_1, C_2)$ and $(D_1, D_2)$,
        respectively, to form new support $(\mathcal{A}^{\ell+1}, \mathcal{B}^{\ell+1})$
        with associated proper path $\Gamma^{\ell+1}$.
```

Fig. 1. Formal Algorithm for finding the Geodesic Tree Path Problem (GTP)

EXPLANATION (2)

- **Input** : two phylogenetic n-trees T = (X, ε, Σ) and T ′ = (X, ε' , Σ′). X = {0,1….,n) is a labeled set of vertices called leaves of degree 1 and ε is the set of interior (non leaf) edges.
- **Output** : A sequence of orthants geodesics between the two phylogenetic n-trees. For all path spaces between the two phylogenetic n -trees the shortest of these path space geodesics will be between the two phylogenetic n-trees.
- **Initialize** : This procedure is implemented once the geodesic is found. The starting proper path chose Γ 0 to be the cone path having support $A^0 = (\varepsilon)$ and

$B^0=(\varepsilon')$. This procedure also satisfies the condition property (P1) and (P2) . The incompatibility graph is defined as graph G(A,B) between sets A ⊆ E and B ⊆ E′ as the bipartite graph between the two phylogenetic n-trees.

- **Iterative Step** : During this step one new orthant is identified that intersects the geodesic and transforms the current path so that it passes through the new orthant. Then each new orthant is identified by finding a minimum weighted vertex cover in a bipartite graph. The incompatibility graph is defined as graph G(A,B) between sets A ⊆ E and B ⊆ E′ as the bipartite graph between the two phylogenetic n-trees.
    - If every minimum weight cover ≥ 1, then the path satisfies (P3), and the current path is the geodesic.
    - If not, we create a new proper path with new supports. We choose any minimum weight vertex covers $C_1 \cup D_1$, ,$C_1 \subset A_i$ and $D_2 \subset B_i$, having weight $\frac{||C_1||^2}{||A_i||^2} + \frac{||D_2||^2}{||B_i||^2} < 1$. We replace A_i and B_i in the set of support pairs by the ordered pairs (C_1, C_2) and (D_1, D_2). With these new support pairs we form a new proper path and proceed.
- ***Minimum weighted vertex cover :*** Uses NP hard complete. (However, for bipartite graphs, in this case it can be solved in O(n^3) . Given an undirected graph G = (A,B) and weighting function defined on the vertex set. The minimum weighted vertex cover is also used to find a vertex set S ⊆ V which has a total weight that is a minimum subject to every edge of G at least one endpoint in S. Described in the Geodesic Tree Path Problem (GTP) algorithm as the graph G = (A, B) with complements C_1 and D_2 that forms a vertex cover for G(A,B) . Every edge of G = (A,B) is part to a vertex of either C_1 and D_2. Furthermore, the minimum weighted vertex cover for G(A,B) has weight $|\square|C_1|\square|^2 + |\square|D_2|\square|^2 < 1$.
- ***Support pairs :*** Are the new orthant added with some subset of the edges in the support pairs and then drop and add the remaining edges to reach the original succeeding orthant that results in a proper path space.

V. IMPLEMENTATION

Our data set will be described in the dataset section, https://cran.r-project.org/web/packages/distory/index.html which we can package to implement our algorithm which we described in the previous section. We use an R library, distory to implement Owen and Provan's algorithm.We will use a visualization tool to display our results. All our code and data will be made available in the github repository:

https://github.com/RagDoll95/TreeSpace

VI. GETTING DATASET

Our dataset is the set of all possible rooted 4-leaf binary trees. The number of rooted binary trees for n taxa is given by

$$(2n-3)!! = \frac{(2n-3)!}{2^{n-2}(n-1)!} \; for \; n \geq 2$$

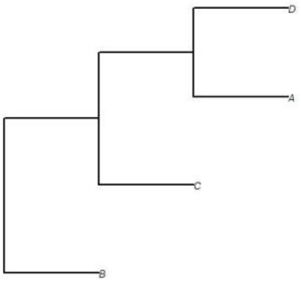
which gives us 15 unrooted trees[5]. This gives us ${}_{15}C_2$ = 105 possible distances[5]. To keep things simple, our process, we will set all edges for every tree to the same length, e = 1. Every tree will have its own orthant then because of its topology will be unique. We use the R library phangorn[6] to generate our trees. The trees themselves will be in Newick format. The labels for our four leaves will be {a, b, c, d}. Our data for all 15 trees can be found in our github repository:

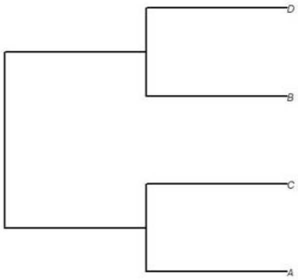
https://github.com/RagDoll95/TreeSpace/tree/master/data.

Figure 2 shows a sample of the first 3 trees. The following are all 4-leaf fifteen rooted binary trees in their respective Newick formats, with all edge lengths equal to 1:

1. (B:1,(C:1,(A:1,D:1):1):1);
2. ((A:1,C:1):1,(B:1,D:1):1);
3. (B:1,(A:1,(C:1,D:1):1):1);
4. (B:1,((A:1,C:1):1,D:1):1);
5. (D:1,(B:1,(A:1,C:1):1):1);
6. ((B:1,C:1):1,(A:1,D:1):1);
7. (A:1,(C:1,(B:1,D:1):1):1);
8. (A:1,(B:1,(C:1,D:1):1):1);
9. (A:1,((B:1,C:1):1,D:1):1);
10. (D:1,(A:1,(B:1,C:1):1):1);
11. (C:1,(B:1,(A:1,D:1):1):1);
12. (C:1,(A:1,(B:1,D:1):1):1);
13. ((A:1,B:1):1,(C:1,D:1):1);
14. (C:1,((A:1,B:1):1,D:1):1);
15. (D:1,(C:1,(A:1,B:1):1):1);

We make a graphical representation of our trees, of which we show three in the paper. To view the rest, consult our github page.


D
A
C
B


D
B
C
A

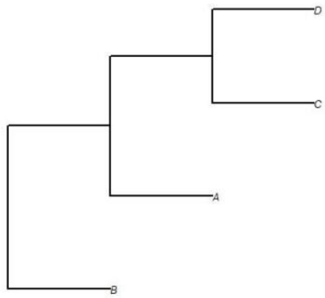


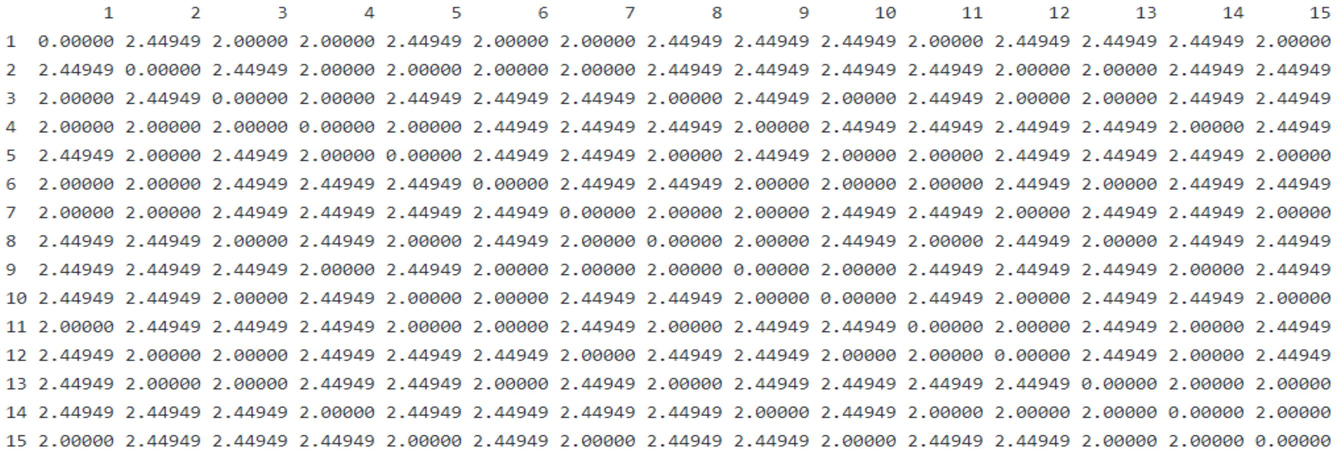
Figure 2: an example of three of the 15 possible 4 leaf trees. We have labeled them 1, 2, and 3 respectively.

VII. METRICS

The metric we get is the geodesic distance between tree pairs. Since every tree has the same edge lengths, the difference in geodesic distances between pairs is caused solely by topology. This will give a good demonstration of BHV space.

VIII. RESULTS

We present the results of our distances below:



```
         1       2       3       4       5       6       7       8       9      10      11      12      13      14      15
1  0.00000 2.44949 2.00000 2.00000 2.44949 2.00000 2.00000 2.44949 2.44949 2.44949 2.00000 2.44949 2.44949 2.44949 2.00000
2  2.44949 0.00000 2.44949 2.00000 2.00000 2.00000 2.00000 2.44949 2.44949 2.44949 2.44949 2.00000 2.00000 2.44949 2.44949
3  2.00000 2.44949 0.00000 2.00000 2.44949 2.44949 2.44949 2.00000 2.44949 2.00000 2.44949 2.00000 2.00000 2.44949 2.44949
4  2.00000 2.00000 2.00000 0.00000 2.00000 2.44949 2.44949 2.44949 2.00000 2.44949 2.44949 2.44949 2.44949 2.00000 2.44949
5  2.44949 2.00000 2.44949 2.00000 0.00000 2.44949 2.44949 2.00000 2.44949 2.00000 2.00000 2.44949 2.44949 2.44949 2.00000
6  2.00000 2.00000 2.44949 2.44949 2.44949 0.00000 2.44949 2.44949 2.00000 2.00000 2.00000 2.44949 2.00000 2.44949 2.44949
7  2.00000 2.00000 2.44949 2.44949 2.44949 2.44949 0.00000 2.00000 2.00000 2.44949 2.44949 2.00000 2.44949 2.44949 2.00000
8  2.44949 2.44949 2.00000 2.44949 2.00000 2.44949 2.00000 0.00000 2.00000 2.44949 2.00000 2.44949 2.00000 2.44949 2.44949
9  2.44949 2.44949 2.44949 2.00000 2.44949 2.00000 2.00000 2.00000 0.00000 2.00000 2.44949 2.44949 2.44949 2.00000 2.44949
10 2.44949 2.44949 2.00000 2.44949 2.00000 2.00000 2.44949 2.44949 2.00000 0.00000 2.44949 2.00000 2.44949 2.44949 2.00000
11 2.00000 2.44949 2.44949 2.44949 2.00000 2.00000 2.44949 2.00000 2.44949 2.44949 0.00000 2.00000 2.44949 2.00000 2.44949
12 2.44949 2.00000 2.00000 2.44949 2.44949 2.44949 2.00000 2.44949 2.44949 2.00000 2.00000 0.00000 2.44949 2.00000 2.44949
13 2.44949 2.00000 2.00000 2.44949 2.44949 2.00000 2.44949 2.00000 2.44949 2.44949 2.44949 2.44949 0.00000 2.00000 2.00000
14 2.44949 2.44949 2.44949 2.00000 2.44949 2.44949 2.44949 2.44949 2.00000 2.44949 2.00000 2.00000 2.00000 0.00000 2.00000
15 2.00000 2.44949 2.44949 2.44949 2.00000 2.44949 2.00000 2.44949 2.44949 2.00000 2.44949 2.44949 2.00000 2.00000 0.00000
```

Figure 3: Distance matrix of all trees

IX. CONCLUSION

We can observe three distinct values for all distances. 0, which is obviously the geodesic distance between any tree and itself, 2.000, and 2.449. Intuitively, it is easy to see why 1 and 3 are closer than 1 and 2 are to each other. The distance can be explained by the number of orthants the geodesic must cross to reach the other trees. This example illustrates intuitively how geodesic distance in BHV space can be a valuable tool to compare trees.

REFERENCES

[1] Louis J Billera, Susan P Holmes, and Karen Vogtmann. Geometry of the space of phylogenetic trees. Advances in Applied Mathematics, , 27(4):733–767, 2001.

[2] Megan Owen and J Scott Provan. A fast algorithm for computing geodesic distances in tree space. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 8(1):2–13, 2011.

[3] Lueg, J., Garba, M.K., Nye, T.M.W., Huckemann, S.F. (2021). Wald Space for Phylogenetic Trees. In: Nielsen, F., Barbaresco, F. (eds) Geometric Science of Information. GSI 2021. Lecture Notes in Computer Science(), vol 12829. Springer, Cham. https://doi.org/10.1007/978-3-030-80209-7_76

[4] Miller, Ezra et al. "Averaging metric phylogenetic trees." *ArXiv* abs/1211.7046 (2012): n. pag.

[5] Felsenstein, Joseph, and Joseph Felenstein. Inferring phylogenies. Vol. 2. Sunderland, MA: Sinauer associates, 2004.

[6] Schliep K.P. 2011. phangorn: phylogenetic analysis in R. Bioinformatics, 27(4) 592-593

[7] Chakerian J, Holmes S (2020). _distory: Distance Between Phylogenetic Histories. R package version 1.4.4, <https://CRAN.R-project.org/package=distory>.