PHYLOGENETIC NETWORKS AND FUNCTIONS THAT RELATE THEM

A Thesis

Presented to

The Williams Honors College of The University of Akron

In Partial Fulfillment

of the Requirements for the Degree

Bachelors of Science

Drew Joseph Scalzo

May, 2020
ABSTRACT

We will be examining the various characteristics of Phylogenetic Networks and functions that take these networks as inputs, and convert them to more complex or simpler structures. Furthermore, we will be examining the nature of functions as they relate to the program NeighborNet, which inputs networks numerically and describes how they interact against multiple types of networks. Finally, we will build upon previous research in this field and attempt to comprise a formula for counting the total number of possible Level-2 Networks.
ACKNOWLEDGEMENTS

My sincere thanks to Dr. Stefan Forcey for his guidance on my research and the many enjoyable hours I have spent learning from him. Many thanks also to Dr. James Cossey and Dr. Kevin Kreider for their time, insightful input, and many interesting conversations. Many thanks to my family for their support and enthusiasm throughout this whole process.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. RELATIONSHIP BETWEEN LEVEL 1 AND LEVEL 2 NETWORKS</td>
<td>6</td>
</tr>
<tr>
<td>III. “WELL-BEHAVED” FUNCTIONS</td>
<td>11</td>
</tr>
<tr>
<td>IV. COUNTING... AND MORE COUNTING</td>
<td>15</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>20</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A visual representation of what we define as a simple Weighted Level 1 Network (N) when compared to a Weighted Level 2 Network (M).</td>
<td>2</td>
</tr>
<tr>
<td>1.3</td>
<td>The diagram depicts the process of taking a Weighted Level 2 Network N and its distance vector (See Figure 2) and putting it into Neighbor Net to obtain the corresponding Split Network. Then, by using our function Lw, we reconstruct our split network into a Weighted Level 1 Network M.</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Observe the corresponding distance vectors of the Level 1 Network R and the Level 2 Network T. Both have the same distance vector, thus illustrating Theorem 2.2.</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>Let X represent the domain of a particular function and let Y represent the same functions codomain. The diagram depicted above shows an injective function, in which every element of X maps to a unique element of Y.</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>Let X represent the domain of a particular function and let Y represent the same functions codomain. The diagram depicted above shows a surjective function, in which every element of Y results from at least one element of X.</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>Note that this Level – 1 Network z can never be the output of Lw.</td>
<td>13</td>
</tr>
<tr>
<td>3.4</td>
<td>Consider the Level – 2 Networks R and T above. By Definition 2.1, we know that since R and T have equivalent distance vectors, they are equivalent networks. That is, when put into Neighbor Net, the resulting Sw(R) = Sw(R). This visualizes Theorem 3.3.</td>
<td>14</td>
</tr>
</tbody>
</table>
4.1 To count this structure, we consider the fact that there are 2 internal diagonals possible (as seen by the dotted lines above) and $\frac{n}{2}$ ways to organize the leaves. Therefore, the total number of Weighted Level 2 Networks with $n = 4$ leaves is $(2)^{\frac{4}{2}} = 6$. . . . . . . . . . . . . . . . . . 16

4.2 For $n = 5$, there exists two structures as constructed above. . . . . . . . 17

4.3 For $n = 6$, there exists six structures as constructed above. . . . . . . . 18
CHAPTER I

INTRODUCTION

The structures we directly address here are split networks and phylogenetic networks. Specifically, the relationship between the functions that connect the two through the program Neighbor Net. We will limit our discussion to the unrooted versions of the structures mentioned above and will focus on the more specific versions of one or both structures, such as: phylogenetic trees, weighted and unweighted level 1 networks, level 2 networks, and split networks.

The definitions used below were translated from [1]. Their paper discussed the relationships between split networks and Level 1 Networks. Towards the end of their paper, they asked is it possible to characterize split systems induced by more complex uprooted networks such as level-2 networks (i.e., networks obtained from level-1 networks by adding a chord to a cycle)? This paper answers that question.

The simplest structure we consider is an (unrooted) phylogenetic tree. Visually, this is a graph with no cycles (a collection of nodes and edges that make a path that returns to the original starting node) and with no nodes of degree 2. The nodes with degree larger than 2 are unlabeled, but the n leaves are labeled bijectively with our n taxa. An (unrooted) phylogenetic network is a simple connected graph with exactly n labeled nodes of degree one, called leaves, all other unlabeled nodes with a
degree of at least three, and where every cycle is of at least length 4. If every node is part of at most one cycle, we call it a Level 1 network. If every node is part of at most two cycles, with an internal diagonal connecting two points of the primary figure, we call it a Level 2 network. It is also known that Level 1 networks, as a set, include Level 0 networks, which are phylogenetic trees; these have no cycles [2].

**Definition 1.1.** We define a Weighted Level 2 Network by any Level 2 Network $N$ with given positive values on every edge of the network. See sample Network $M$ in Figure 1 for an example.

![Figure 1.1: A visual representation of what we define as a simple Weighted Level 1 Network (N) when compared to a Weighted Level 2 Network (M).](image-url)
We can take advantage of a program known as Neighbor Net (NN) [3] to convert a Weighted Level 2 network to a Level 1 network. We first formulate the distance vector for the Level 2 network (denoted by $d_N$), where we let $d_N$ be the distance vector on the leaves of $N$ defined by $d_N(i,j)$ equal to the least sum of weights along a path between leaves $i$ and $j$; see Figure 2 for a visual of this process. Neighbor Net is used to convert the network into a weighted split network, a specially connected simple graph where each split is represented by a set of parallel edges, each split has an assigned weight, and is a minimal cut of the graph. Finally, there is a simple way to associate a split network $s$ to a specific Level 1 network through $L(s)$. We make this precise in Chapter 2.

**Definition 1.2.** Construct the network $L(s)$ as follows: begin with a split network diagram of $s$ and consider the diagram as a planar drawing of its underlying planar graph, with leaves on the exterior. Then 1) delete all the edges that are not adjacent to the exterior of that graph, and 2) smooth away any resulting degree 2 nodes. [4]

**Definition 1.3.** We construct the network $Lw(s)$ by following the same steps as the network $L(s)$ but then summing up the split lengths to get the edge lengths. We will illustrate this concept in Figure 3.
Figure 1.2: Consider the Weighted Level – 2 Network $N$ above. The distance vector, denoted $d_N$, is equal to the least sum of weights between two leaves (in bold). The distance vector for $N$ is...

$$d_N = < d_{12}, d_{13}, d_{14}, d_{15}, d_{16}, d_{23}, d_{24}, d_{25}, d_{26}, d_{34}, d_{35}, d_{36}, d_{45}, d_{46}, d_{56} >$$

$$d_N = < 4, 7, 5, 8, 7, 5, 7, 10, 9, 7, 13, 12, 10, 9, 3 >$$

Note that $d_{12}$, for example, refers to the shortest distance between the leaves 1 and 2.

We will now illustrate Definition 1.3. To begin, refer to Figure 1.3. Now consider the edge of length 2.5 (boxed) in the cycle, between the leaves 3 and 4 in Figure 3 (next page). We calculated 2.5 by adding 2 and 0.5 from Sw(NN), which are the lengths of the splits separating the leaves 3 and 4. Note that any generic network that has no assigned weights on its edges is considered to be unweighted, or written simply without the word weighted preceding the name.
Figure 1.3: The diagram depicts the process of taking a Weighted Level 2 Network N and its distance vector (See Figure 2) and putting it into Neighbor Net to obtain the corresponding Split Network. Then, by using our function $L_w$, we reconstruct our split network into a Weighted Level 1 Network $M$. 
CHAPTER II

RELATIONSHIP BETWEEN LEVEL 1 AND LEVEL 2 NETWORKS

Recall that a Level 2 Network is a simple graph where every node is a part of at most 2 cycles and which contains an internal diagonal connecting two points of the primary figure. To obtain the distance vector for any weighted Level 2 network, we once again compile the shortest distances between all the possible pairs of leaves, and then combine those numerical distances into a vector. However, when considering the length of the internal diagonal of a Level – 2 network, which we will from here on out call the internal edge, it would make sense that if the length exceeds a certain numerical barrier, it would no longer serve as an optimal (or minimal) route between two nodes. We will discuss this scenario later on.

A crucial step in the Neighbor Net system includes the process of taking a Weighted Level K Network (where $K > 1$) and converting it to its weighted split network form, but first it is important to note what we mean by a split network. In general, a split system is any collection of splits which contains all the possible number of ways a network may partition itself into two parts. A split network is a graphical representation of a split system, and more specifically, a weighted split network is a graphical representation of a weighted network. Each split is represented by a set of parallel lines which is the minimal cut of the graph; that is, removing that
set of edges would separate the graph into two components, whose respective nodes are part of the split. We utilize this form to garner the edge lengths that form the eventual Level 1 Network. See Figure 3 for an example of a weighted split network.

**Definition 2.1.** For a Weighted Level 2 Network $N$, we define $Sw(N)$ to be the weighted split network found by $Sw(N) = NN(d_N)$.

**Lemma 2.1.** If $d_N$ is the distance vector on the leaves of $N$ defined by $d_N(i,j)$ equal to the least sum of weights along a path between leaves $i$ and $j$, then there is a unique circular weighted split system $s = Sw(N)$ which has the same associated distance vector. That is, $d_N = d_s$.

First we show that $d_N$ obeys the Kalmanson condition: there exists a circular ordering of $[n]$ such that for all $1 \leq i < j < k < l \leq n$ in that ordering,

$$\max[d_N(i,j) + d_N(k,l), d_N(j,k) + d_N(i,l)] \leq d_N(i,k) + d_N(j,l)$$

The circular ordering that meets our specifications is just any choice of one of the circular orderings consistent with $N$. Our network $N$ is planar, so the edges are drawn with no crossings. The two paths involved on the right hand side of the condition intersect each other. Then since the leaves are on the exterior, the four paths involved on the left hand side of the condition are each bounded above in length by a path made by following first one intersecting path and then the other, (switching at the crossroads, after their shared portion). Two paths in a sum on the left hand side of the condition can at most use exactly all of both the crossing paths, so that the inequality is guaranteed.
It is well known that for any Kalmanson metric $d_N$ there exists a unique weighted split system $s$ whose weighting gives that metric: $d_N = ds$. To actually calculate this split system, the algorithm neighbor-net can be used; since it is guaranteed to return the unique answer for any Kalmanson metric. [4]

**Theorem 2.1.** For every weighted Level 2 network $N$, there exists some (not unique) weighted Level 1 network $M$ such that $d_N = d_M$.

By Lemma 2.1., we know that for every Level 2 network, there exists a unique circular weighted split system $Sw(N)$. Neighbor-net assigns each split of $Sw(N)$ a positive value, since when splits are assigned weight = 0 this system can be equated to the system minus those splits. Given these weights, we can use our function $Lw$ to obtain a weighted Level 1 network $M$. The weight of an edge in $M = Lw (Sw(N))$ is the sum of the weights of the splits corresponding to the edge. Then the function $Lw$ creates the Level – 1 network $M$, described by $M = Lw (Sw(N))$ that corresponds to the original Level – 2 network. Now we have $d_N = ds$ and $d_M = ds$ by Lemma 2.1. That is, $d_N = d_M$. (See Figure 3)

**Theorem 2.2.** For every weighted Level – 1 network $M$, there exists some (not unique) weighted Level – 2 network $N$ such that $d_M = d_N$.

Consider a Level 1 network $M$ with positive values for its edges and a Level 2 network $N$ with the same positive values for its edges, but a large positive value for its internal chord. Let $d_N$ be the distance vector on the leaves of $N$ defined by $d_N(i,j)$ equal to the least sum of weights along a path between leaves $i$ and $j$. Note that the internal chord of the Level 2 network will never get used, and thus, will not
change the distance values for its network. Thus, both networks will result in the same distance vector $d_M = d_N$ (see Figure 2.1).

Figure 2.1: Observe the corresponding distance vectors of the Level 1 Network $R$ and the Level 2 Network $T$. Both have the same distance vector, thus illustrating Theorem 2.2.

**Definition 2.2.** Two Level 2 networks are equivalent, $N \equiv N'$, if $d_N = d_{N'}$. (Clearly reflexive, transitive, and symmetric properties follow directly from equality of vectors)

We consider this relationship to describe a deeper relationship between Level 1 and Level 2 Networks. If we consider Theorems 2.1 and 2.2, then we may derive a hypothesis about their equivalence. We know that every Weighted Level 1 Network
corresponds to a (non-unique) Weighted Level 2 Network, so we can deduce that
the two networks are representatives of a larger set of networks, all related by their
corresponding distance vector. That is, the equivalent class containing these networks
must contain a component from Level 1 Networks and another from Level 2 Networks.

**Result 2.1.** *Each equivalence class has both a representative that is Level 2
and a representative that is Level 1.*

Recall we have defined an equivalence relation on network s by setting equiva-
 lent networks which have the same distance vector. First, we will show that every
Level – 2 network is equivalent to some Level – 1 network. By Theorem 2.1., we
know that for every Level – 2 network, there exists some weighted Level – 1 network
corresponding to the original Level – 2 network. That is, every Level – 2 network
can be converted to some Level – 1 network. Second, we will show that every Level
– 1 Network is equivalent to some Level – 2 Network. Given a Level – 1 Network,
we insert a chord that has a weight larger than any distance in the distance vector.
By Theorem 1.2., since both networks have equal values for their external edges and
the internal chord of the Level – 2 network is large enough to have no effect on the
distance vector, both networks will result in the same distance vector. Thus, any
Level – 2 network is equivalent to some Level – 1 network.
CHAPTER III

“WELL-BEHAVED” FUNCTIONS

NN refers to the Neighbor Net program.

Definition 3.1. We define a function to be injective, or “one-to-one”, if every input equates to a unique output. In other words, a function is injective if every element of its codomain (set of outputs) maps to at most one element of its domain (set of inputs). (see Figure 3.1)

Figure 3.1: Let X represent the domain of a particular function and let Y represent the same functions codomain. The diagram depicted above shows an injective function, in which every element of X maps to a unique element of Y.
Definition 3.2. We define a function to be surjective, or “onto”, if every element of its codomain (set of outputs) results from at least one input from its domain (set of inputs). (see Figure 3.2)

Figure 3.2: Let X represent the domain of a particular function and let Y represent the same function's codomain. The diagram depicted above shows a surjective function, in which every element of Y results from at least one element of X.

Theorem 3.1. The function Sw is surjective.

We will show for every weighted split network y, there exists a Level 2 weighted network x, such that Sw (x) = y. For y, we first find Lw (y) and let x be any Level 2 network that is equivalent to Lw (y). By Theorem 2.2., we know dL(y) = dy and dx = dL(y). Since NN(dy) = y (NN always give the unique weighted split network for any Kalmanson metric d [[5]]) and dx = dy, it follows that Sw (x) = NN(dx) = NN(dy) = y.
**Theorem 3.2.** The function $L_w$ is not surjective.

We will show there exists a weighted Level 1 network $z$, such that there does not exist a split network $y$, where $L_w(y) = z$. Suppose to get a contradiction, let a weighted Level – 1 network $z$ have one side of a quadrilateral be length 100 and all other sides be less than 5. Let $L_w(y) = z$, then $y$ has a split length 100, but that implies $L_w(y)$ has two side lengths of 100, a contradiction. Thus, $L_w(y) \neq z$.

![Figure 3.3: Note that this Level – 1 Network $z$ can never be the output of $L_w$.](image)

**Theorem 3.3.** The function $S_w$ is not injective.

Consider any two equivalent Level 2 networks with different edge lengths. Both networks will result in the same $S_w(x) = S_w(y)$. Thus, $S_w$ is not injective. (See Figure 9)
Figure 3.4: Consider the Level – 2 Networks R and T above. By Definition 2.1, we know that since R and T have equivalent distance vectors, they are equivalent networks. That is, when put into Neighbor Net, the resulting Sw(R) = Sw(R). This visualizes Theorem 3.3.

**Theorem 3.4.** The function Lw is injective.

We will show that given a Level 1 network z, a split network x, and \( z = Lw(x) \), we can define \( Lw^{-1} \) so that \( Lw^{-1}(z) = x \). Let \( Lw^{-1}(z) = Sw(z) \). Observe that by definition, \( Sw(z) = NN(dz) = NN(dx) = x \).
We would like to know how many unweighted Level – 2 Networks exist with n leaves and k bridges. We first define a bridge to be an edge that when deleted, disconnects a network (breaks it cycle). This suggests that we may combine multiple structures to form a larger phylogenetic network with n leaves. Refer to Figure 4.1 for an example of two Level – 1 Networks connected by a bridge, but which consist of different attached structures.

It is important to note that if we limit a network to having a cycle greater than 3, then it is possible to find the maximum number of possible connected structures with a specific number of leaves. By limiting ourselves to Level – 1 Networks, previous work in this field notes that the number of possible Level – 1 Networks with n leaves can be given by [4]. Again, this section will discuss the findings of counting Level – 2 Networks and compare those results to the counting done in previous research for Level – 1 Networks. Once again, we will limit ourselves to networks of cycle greater than 3 and leaves greater than 4.

We will first consider structures with 4 leaves (n=4). There is only one structure and it is constructed follows:
Figure 4.1: To count this structure, we consider the fact that there are 2 internal
diagonals possible (as seen by the dotted lines above) and $\frac{3!}{2}$ ways to organize the
leaves. Therefore, the total number of Weighted Level 2 Networks with $n = 4$ leaves
is $(2) \frac{3!}{2} = 6$

Again, we count these by looking at each picture individually. Then, we count
the total number of possible internal diagonals of the structure and the multiple
by the possible number of cycles (the number of ways to reorganize the leaves of
the structure). This becomes complicated when we consider bridges, so to include
this factor, we simply divide $2k$, where $k$ is the number of bridges, so that we may
eliminate the resulting combinations when rotating the structure about the bridge.
Also, if symmetry occurs between a pair of branches, we again divide the number of
possible internal diagonals by 2, to address the possibility of overcounting the total
number of structures. Thus, we obtain the following.
This procedure can be followed for \( n = 5 \).

![Figure 4.2](image)

Figure 4.2: For \( n = 5 \), there exists two structures as constructed above.

The counting for each structure in Figure 10 is as follows...

\[
\frac{5(2) \cdot 4!}{2 \cdot 2} = 60
\]

\[
\frac{4(1) \cdot 5! \cdot 1}{2 \cdot 2 \cdot 2} = 60
\]

Total number of combinations is \( = 60 + 60 = 120 \).

As stated previously, we counted these structures by first noticing there were a possible 5 internal chords for one structure, and a possible 2 internal chords for the other. We then multiplied by the number of ways to rearrange the leaves of the structure counterclockwise, which is why we divided by 2. Finally, if there was a bridge connecting any components of the structure, we divided by 2, which can be observed in the second calculation for Figure 4.2.
Similarly for \( n = 6 \).

\[ \begin{array}{c}
\text{(a)} \quad \text{(b)} \quad \text{(c)} \\
\text{(d)} \quad \text{(e)} \quad \text{(f)}
\end{array} \]

Figure 4.3: For \( n = 6 \), there exists six structures as constructed above.

The counting for each structure is as follows (from a to f)...

(a) \( \frac{6(3)}{2} \frac{5!}{2} = 540 \)

(b) \( (2)(2) \frac{4(1)}{2} \frac{6!}{2} \frac{1}{2} \frac{1}{2} = 720 \)

(c) \( \frac{4(1)}{2} \frac{6!}{2} \frac{1}{2} \frac{1}{2} = 90 \)

(d) \( \frac{5(2)}{2} \frac{6!}{2} \frac{1}{2} = 900 \)

(e) \( \frac{4(1)}{2} \frac{6!}{2} \frac{1}{2} \frac{1}{2} = 180 \)

(f) \( \frac{4(1)}{2} (6!) \frac{1}{4} = 360 \)

Total number of combinations is \( = 540 + 720 + 90 + 900 + 180 + 360 = 2790 \).

Notice for (f), reading the labels clockwise is not equivalent to reading them counterclockwise due the tree structures. This means we just consider 6! and not \( \frac{6!}{2} \).
To recap, we would like to know how many unweighted Level – 2 networks exist with \( n \) leaves and \( k \) bridges. However, as seen above, in order to fully count the total number of structures, we must know all the types of structures that can be drawn and count each of those individually. Since every structure is composed differently, in terms of the number of bridges and the appearance of symmetry, it is extremely difficult to generalize a formula for \( n \) leaves. The best method we have is to physically draw out all the structures and consider different factors for each one. This will give us the total number of possible Level - 2 Phylogenetic Networks with \( n \) leaves.
BIBLIOGRAPHY


