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

PHYLOGENETIC NETWORKS AND FUNCTIONS THAT RELATE THEM

Drew Joseph Scalzo

Thesis

Approved:

____
Advisor
Dr. Stefan Forcey

____
Faculty Reader
Dr. James Cossey

____
Faculty Reader
Dr. Kevin Kreider

____
Department Chair
Dr. Kevin Kreider

Accepted:

____
Dean of the College of Arts and Sciences
Dr. Linda Subich

____
Dean of the Williams Honors College
Dr. Dane Quinn

____
Date

ABSTRACT

Phylogenetic Networks are defined to be simple connected graphs with exactly n labeled nodes of degree one, called leaves, and where all other unlabeled nodes have a degree of at least three. These structures assist us with analyzing ancestral history, and its close relative - phylogenetic trees - garner the same visualization, but without the graph being forced to be connected. In this paper, 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 unweighted binary, triangle free, 2-nested 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 for his ever-so-clever jokes and teachings and Dr. Kevin Kreider for his mentoring and guidance throughout my undergraduate endeavour. I would also like to thank my family for their support and enthusiasm throughout this whole process.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION TO PHYLOGENETIC NETWORKS

The structures we directly address here are split networks and phylogenetic networks. Specifically, we consider 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 1-nested networks, 2-nested networks, and split networks.

The definitions used below were translated from [1]. Their paper discussed the relationships between split networks and 1-nested networks. Towards the end of their paper, they asked: is it possible to characterize split systems induced by more complex uprooted networks such as 2-nested networks (i.e., networks obtained from 1-nested 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, and where all other unlabeled nodes have a degree of at least three. If every cycle is of at least length 4 and every edge is part of at most one cycle, we call it a 1-nested network. If every edge is part of at most two cycles (with an internal chord connecting two points of the primary figure), we call it a 2-nested network. It is also known that 1-nested networks, as a set, include 0-nested networks, which are phylogenetic trees; these have no cycles. [2]

Definition 1.1. *We define a weighted 2-nested network by any 2-nested network N with given positive values on every edge of the network. See sample network M in Figure 1.1 for an example.*

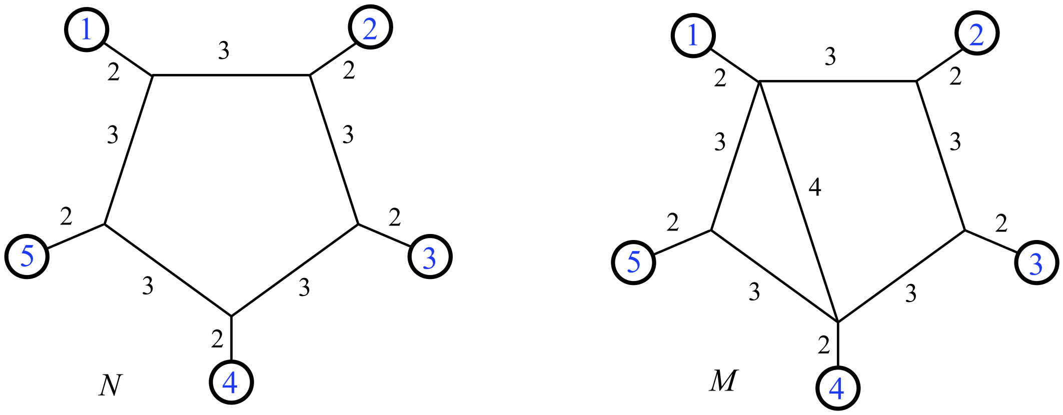

Figure 1.1: A visual representation of what we define as a weighted 1-nested network N when compared to a weighted 2-nested network M.

We can take advantage of a program known as Neighbor Net (NN) [3] to convert a weighted 2-nested network to a 1-nested network. We first formulate the distance vector for the 2-nested network (denoted by $\mathbf{d}_N$), where we let $\mathbf{d}_N$ be the distance vector on the leaves of N defined by $\mathbf{d}_N(i, j)$ equal to the least sum of weights along a path between leaves i and j; see Figure 1.2 for a visual of this process. Neighbor Net is then used to convert the network into a weighted split network, a special 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 take a split network s to a specific 1-nested 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 $L_W(s)$ 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.*

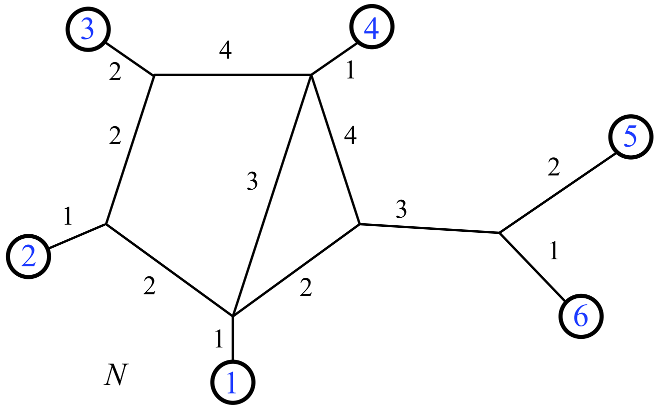

Figure 1.2: Consider the weighted 2-nested network N above. The distance vector, denoted $\mathbf{d}_N$, is equal to the least sum of weights between two leaves (in bold). The distance vector for N is...

$$\mathbf{d}_N =< \mathbf{d}_{12}, \mathbf{d}_{13}, \mathbf{d}_{14}, \mathbf{d}_{15}, \mathbf{d}_{16}, \mathbf{d}_{23}, \mathbf{d}_{24}, \mathbf{d}_{25}, \mathbf{d}_{26}, \mathbf{d}_{34}, \mathbf{d}_{35}, \mathbf{d}_{36}, \mathbf{d}_{45}, \mathbf{d}_{46}, \mathbf{d}_{56} >$$

$$\mathbf{d}_N =< 4, 7, 5, 8, 7, 5, 7, 10, 9, 7, 13, 12, 10, 9, 3 >$$

Note that $\mathbf{d}_{12}$, 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 1.3 (next page). We calculated 2.5 by adding 2 and 0.5 from S_W(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.

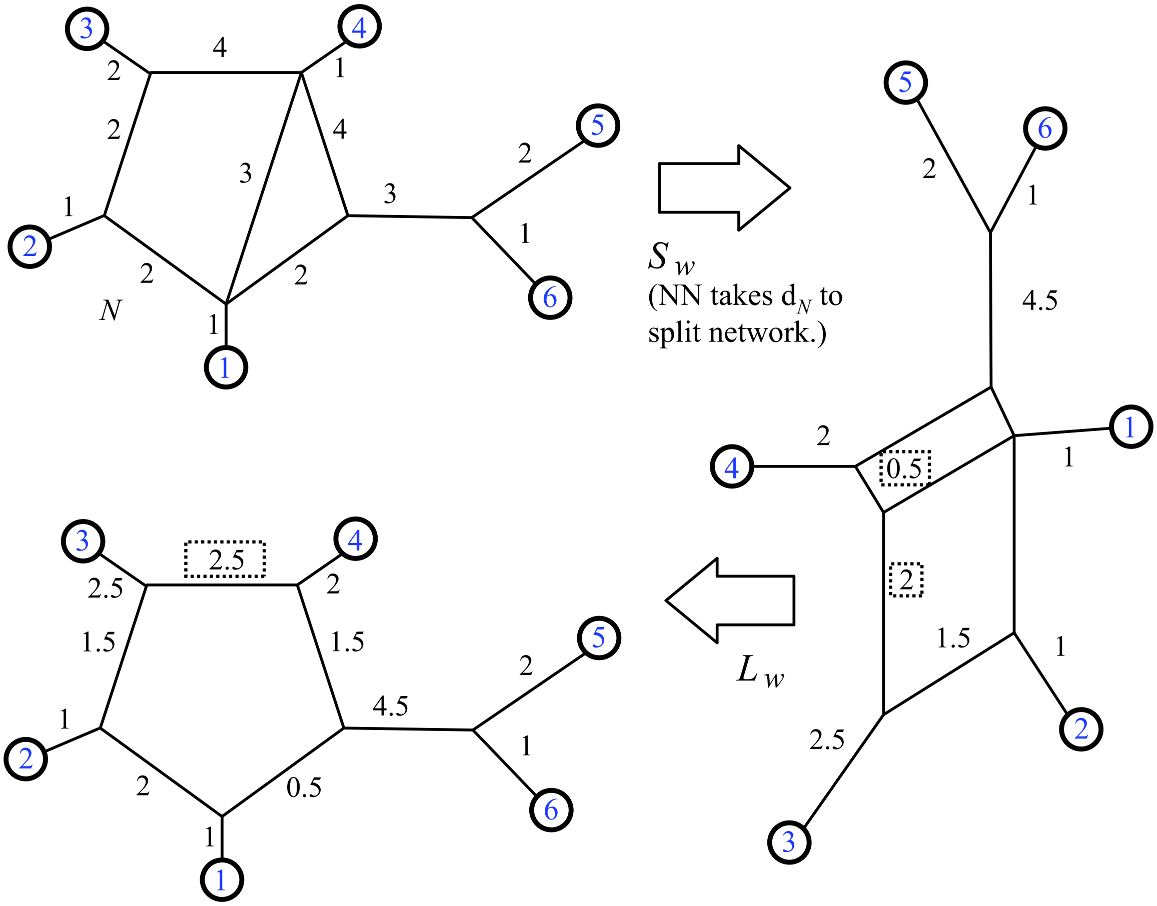

Figure 1.3: The diagram depicts the process of taking a weighted 2-nested network N and its distance vector (See Figure 2) and putting it into Neighbor Net to obtain the corresponding split network. Then, by using our function L_w, we reconstruct our split network into a weighted 1-nested network.

CHAPTER II

RELATIONSHIP BETWEEN 1-NESTED AND 2-NESTED NETWORKS

Recall that a 2-nested network is a simple graph where every edge is a part of at most 2 cycles. To obtain the distance vector for any weighted 2-nested 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 2-nested network, 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 K-nested 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 1-nested network. See Figure 1.3 for an example of a weighted split network.

Definition 2.1. *For a weighted 2-nested network N, we define S_W(N) to be the weighted split network found by S_W(N) = NN($\mathbf{d}_N$).*

Lemma 2.1. *If $\mathbf{d}_N$ is the distance vector on the leaves of N defined by $\mathbf{d}_N$(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 = S_W(N) which has the same associated distance vector. That is, $\mathbf{d}_N = \mathbf{d}_s$.*

Proof: First we show that $\mathbf{d}_N$ 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,

$$\max\left[\mathbf{d}_N(i,j) + \mathbf{d}_N(k,l), \mathbf{d}_N(j,k) + \mathbf{d}_N(i,l) \leq \mathbf{d}_N(i,k) + \mathbf{d}_N(j,l)\right]$$

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 $\mathbf{d}_N$ there exists a unique weighted split system s whose weighting gives that metric: $\mathbf{d}_N = \mathbf{d}_s$. 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 2-nested network N, there exists some (not unique) weighted 1-nested network M such that* $\mathbf{d}_N = \boldsymbol{d}_M$.

<u>*Proof:*</u> By Lemma 2.1., we know that for every 2-nested network, there exists a unique circular weighted split system S_W(N). Neighbor-net assigns each split of S_W(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 L_W to obtain a 1-nested network M. The weight of an edge in $M = L_W\ (S_W(N))$ is the sum of the weights of the splits corresponding to the edge. Then the function L_W creates the 1-nested network M, described by $M = L_W\ (S_W(N))$ that corresponds to the original 2-nested network. Now we have $\mathbf{d}_N = \mathbf{d}_s$ and $\mathbf{d}_M = \mathbf{d}_s$ by Lemma 2.1. That is, $\mathbf{d}_N = \mathbf{d}_M$. (See Figure 1.3) ■

Theorem 2.2. *For every weighted 1-nested network M, there exists some (not unique) weighted 2-nested network N such that* $\mathbf{d}_M = \mathbf{d}_N$.

Proof: Consider a 1-nested network M with positive values for its edges and a 2-nested network N with the same positive values for its edges, but a large positive value for its internal chord. Let $\mathbf{d}_N$ be the distance vector on the leaves of N defined by $\mathbf{d}_N(i,j)$ equal to the least sum of weights along a path between leaves i and j. Note that the internal chord of the 2-nested network will never get used, and thus, will not change the distance values for its network. Therefore, both networks will result in the same distance vector $\mathbf{d}_M = \mathbf{d}_N$ (see Figure 2.1). ■

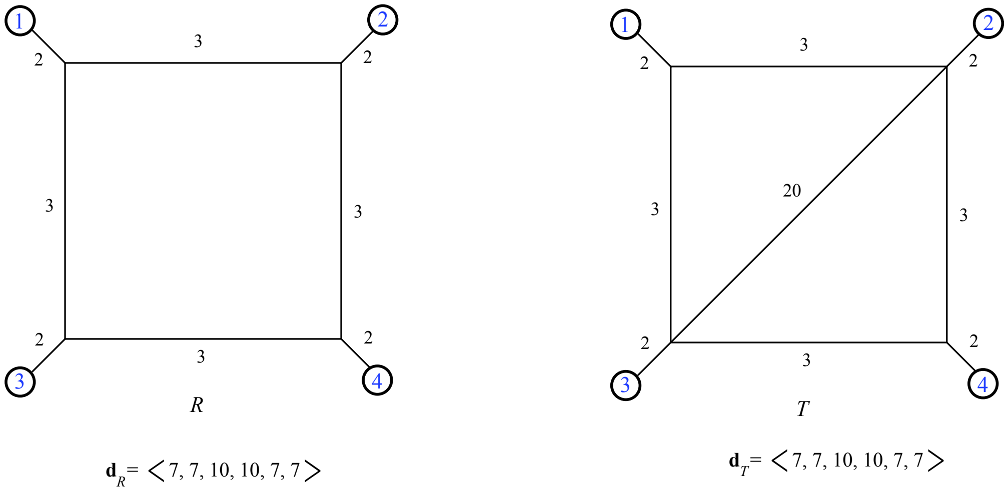
1
2
3
4
2
3
3
3
3
2
2
2
R
1
2
3
4
20
T
$\mathbf{d}_R = \langle 7, 7, 10, 10, 7, 7 \rangle$
$\mathbf{d}_T = \langle 7, 7, 10, 10, 7, 7 \rangle$

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

Definition 2.2. *Two 2-nested networks are equivalent, $N \sim N'$, if $\mathbf{d}_N = \mathbf{d}_{N'}$. (Clearly reflexive, transitive, and symmetric properties follow directly from equality of vectors)*

We consider this relationship to describe a deeper relationship between 1-nested and 2-nested networks. If we consider Theorems 2.1 and 2.2, then we may derive a hypothesis about their equivalence. We know that every weighted 1-nested network corresponds to a (non-unique) weighted 2-nested 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 1-nested networks and another from 2-nested networks.

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

Proof: Recall we have defined an equivalence relation on network s by setting equivalent networks which have the same distance vector. First, we will show that every 2-nested network is equivalent to some 1-nested network. By Theorem 2.1., we know that for every 2-nested network, there exists some weighted 1-nested network corresponding to the original 2-nested network. That is, every 2-nested network can be made into some 1-nested network. Second, we will show that every 1-nested network is equivalent to some 2-nested network. Given a 1-nested network, we insert a chord that has a weight larger than any distance in the distance vector. By Theorem 2.2., since both networks have equal values for their external edges and the internal chord of the 2-nested network is large enough to have no effect on the distance vector, both networks will result in the same distance vector. So, any 1-nested network can be made into some 2-nested network. Therefore, any equivalence class formed by 1-nested and 2-nested networks has both a representative that is 2-nested and a representative that is 1-nested. ■

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).*

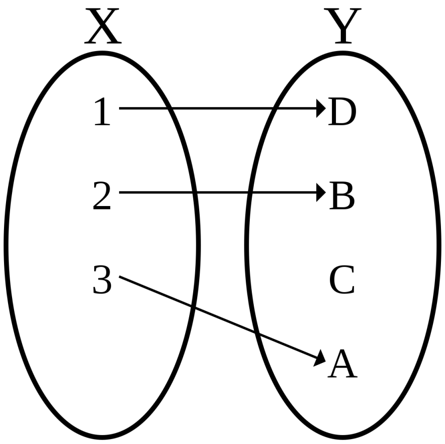

Figure 3.1: Let X represent the domain of a particular function and let Y represent the same functions codomain. The diagram depicted above shows an injective function, in which every element of X maps to a unique element of Y.

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).*

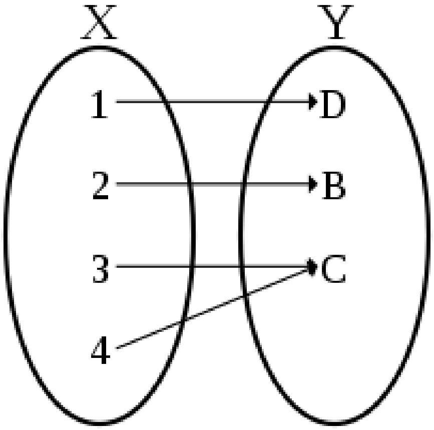

Figure 3.2: Let X represent the domain of a particular function and let Y represent the same functions codomain. The diagram depicted above shows a surjective function, in which every element of Y results from at least one element of X.

Theorem 3.1. *The function S_W is surjective.*

<u>Proof:</u> We will show for every weighted split network y, there exists a 2-nested weighted network x, such that $S_W(x) = y$. For y, we first find $L_W(y)$ and let x be any 2-nested network that is equivalent to $L_W(y)$. By Theorem 2.2., we know $\mathbf{d}_{L(y)} = \mathbf{d}_y$ and $\mathbf{d}_x = \mathbf{d}_{L(y)}$. Since $\mathrm{NN}(\mathbf{d}_y) = y$ (NN always give the unique weighted split network for any Kalmanson metric $\mathbf{d}_{([5])}$) and $\mathbf{d}_x = \mathbf{d}_y$, it follows that $S_W(x) = \mathrm{NN}(\mathbf{d}_x) = NN(\mathbf{d}_y) = y$. ■

Theorem 3.2. *The function L_W is not surjective.*

<u>*Proof:*</u> We will show there exists a weighted 1-nested 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 1-nested 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 clear contradiction. Thus, $L_W(y) \neq z$. ■

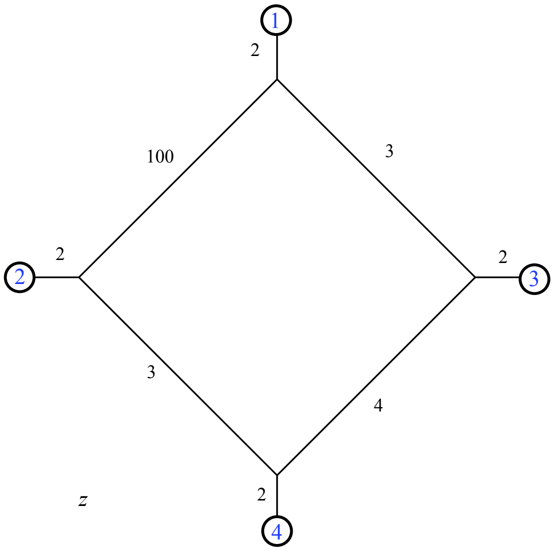

Figure 3.3: Note that this 1-nested network z can never be the output of L_W.

Theorem 3.3. *The function S_W is not injective.*

<u>*Proof:*</u> Consider any two equivalent 2-nested 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 3.4) ■

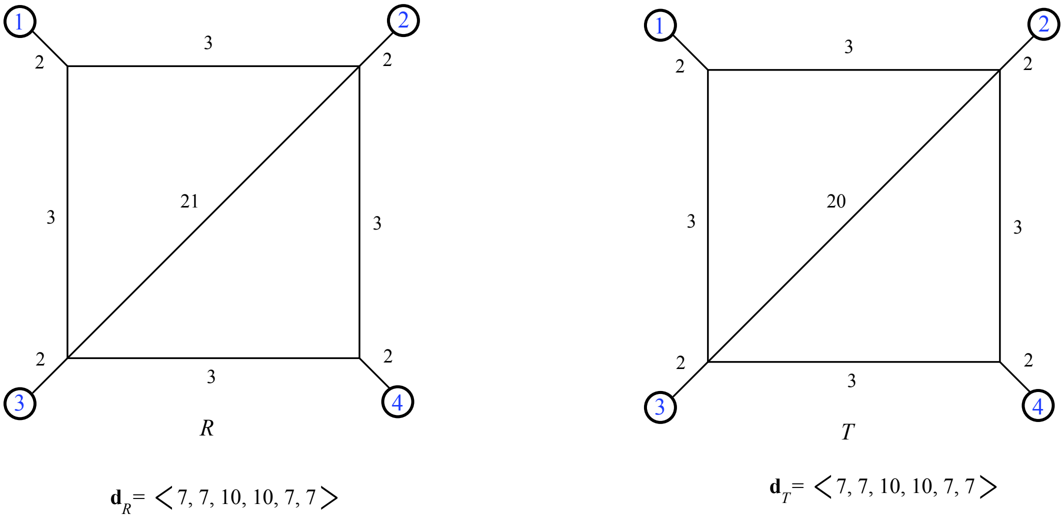
1
2
3
4
3
2
2
3
21
3
2
2
3
R
$\mathbf{d}_R = \langle 7, 7, 10, 10, 7, 7 \rangle$
1
2
3
4
3
2
2
3
20
3
2
2
3
T
$\mathbf{d}_T = \langle 7, 7, 10, 10, 7, 7 \rangle$

Figure 3.4: Consider the 2-nested networks R and T above. By Definition 2.1, we know that since R and T have equivalent distance vectors, so by definition they are equivalent networks. That is, when we put their corresponding distance vectors into Neighbor Net, the two resulting split networks are the same: $S_W(R) = S_W(R)$.

Theorem 3.4. *The function L_W is injective.*

Proof: We will show that given a 1-nested network z, a split network x, and $z = L_W(x)$, we can define L_W^{-1} so that $L_W^{-1}(z) = x$. Let $L_W^{-1}(z) = S_W(z)$. Observe that by definition, $S_W(z) = \mathrm{NN}(\mathbf{d}_z) = \mathrm{NN}(\mathbf{d}_x) = x$. ■

CHAPTER IV

COUNTING... AND MORE COUNTING

We would like to know how many unweighted binary, triangle free, 2-nested 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 allows us to combine multiple structures to form a larger phylogenetic network with n leaves.

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 1-nested networks, previous work in this field notes that the number of possible 1-nested networks with n leaves can be given by [4]. Again, this section will discuss the findings of counting unweighted binary, triangle free, 2-nested networks and then compare those results to the counting done in previous research for 1-nested 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:

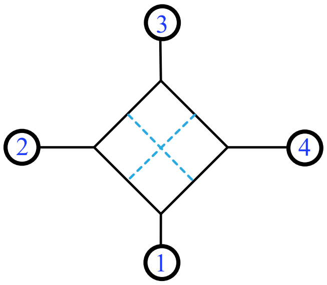

Figure 4.1: To count this structure, we consider the fact that there are 2 internal diagonals possible (as seen by the dotted lines above) and $\frac{3!}{2}$ ways to organize the leaves. Therefore, the total number of weighted 2-nested networks with $n = 4$ leaves is $(2)\frac{3!}{2} = 6$

Again, we count these by looking at each picture individually. Then, we count the total number of possible internal diagonals of the structure and then multiply 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 by $2k$, 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.

This procedure can be followed for $n = 5$.

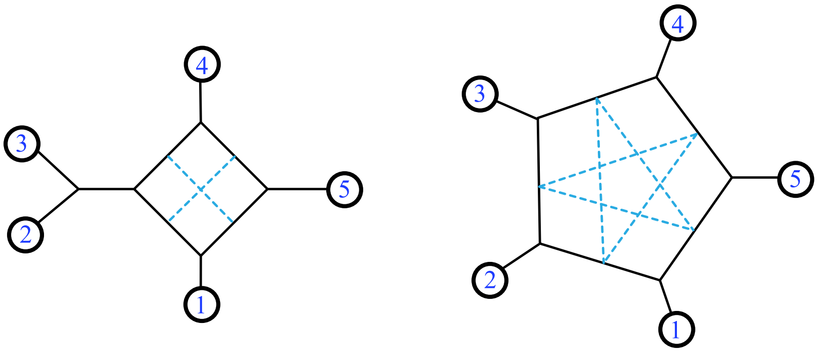

Figure 4.2: For $n = 5$, there exists two structures as constructed above.

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

$$\frac{5(2)}{2}\frac{4!}{2} = 60$$

$$\frac{4(1)}{2}\frac{5!}{2}\frac{1}{2} = 60$$

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 ($n!$) counterclockwise. However, rearranging the leaves clockwise and counterclockwise yield the same rearrangement, so we must then divide by 2 to eliminate half of the arrangements garnered from the counting of those leaves. 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 $n = 6$.

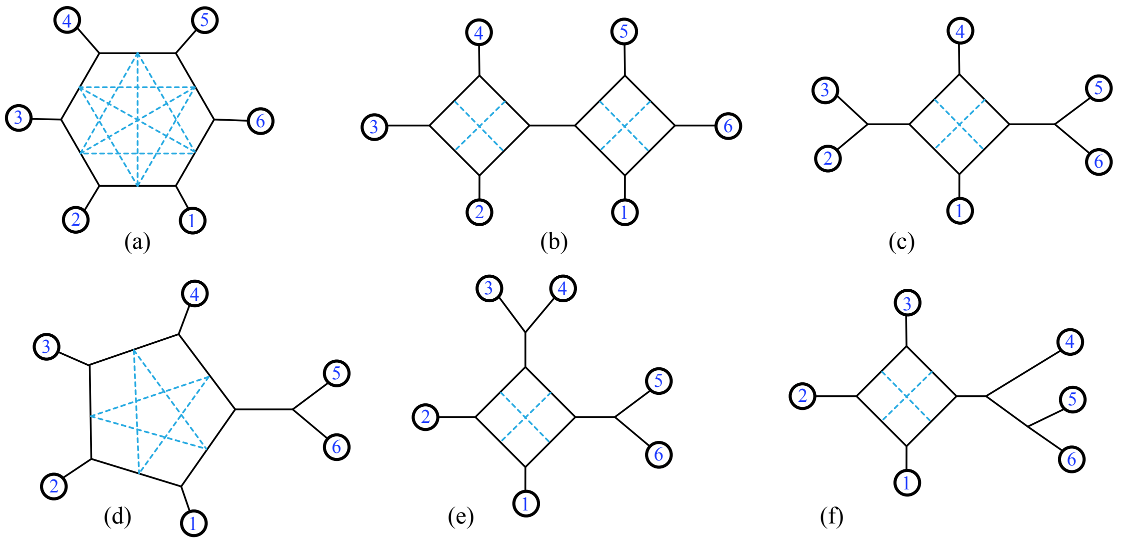

Figure 4.3: For $n = 6$, there exists six structures as constructed above.

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

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

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

(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 2-nested 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 unweighted binary, triangle free, 2-nested phylogenetic networks with n leaves.

BIBLIOGRAPHY

[1] P. Gambette, K. T. Huber, and G. E. Scholz. Uprooted phylogenetic networks. *Bull. Math. Biol.*, 79(9):2022–2048, 2017.

[2] Stefan Forcey and Drew Scalzo. Galois connections and duality between phylogenetic network spaces and polytopes. 2019.

[3] Dan Levy and Lior Pachter. The neighbor-net algorithm. *Adv. in Appl. Math.*, 47(2):240–258, 2011.

[4] Cassandra Durell and Stefan Forcey. Level – 1 phylogenetic networks and their balanced minimum evolution polytopes. *arXiv:1905.09160 [math.CO]*, 2019.

[5] Mike Steel. *Phylogeny—discrete and random processes in evolution*, volume 89 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2016.