2021-12-13

# Modeling a Human Family Network

Rebecca Jo Flores
*Brigham Young University*

# Modeling a Human Family Network

Rebecca Jo Flores

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Benjamin Webb, Chair
Emily Evans
Mark Kempton

Department of Mathematics

Brigham Young University

# ABSTRACT

## Modeling a Human Family Network

Rebecca Jo Flores
Department of Mathematics, BYU
Master of Science

We propose a model that generates a family network based on real-world family network data. We use this model to study the extent to which distances to union and the number of children characterize family networks. To determine how accurate our model is we use persistent homology to identify and compare the structure of our modeled family networks to real-world family networks. To accomplish this, we introduce the notion of a network's persistence curve, which encodes the network's set of persistence intervals. Using the bottleneck distance allows us to measure the difference in the homological structure between any pair of networks. We also study how the distribution of distance to union and the distribution of children build family networks. What we find is that these two features of distance to union and number of children allow us to fairly accurately recreate family networks at least at the level of their persistent homology.

Keywords: family networks, persistent homology, persistence curves, bottleneck distance

# Acknowledgements

Thank you to Dr. Nick Callor, Taylor Gledhill, Abigail Jenkins, Dr. Robert Snellman, and Raelynn Wonnacott for all your help through this whole process. Your collaboration and insights were invaluable. Thanks to my family for supporting me and believing in me. Most of all, thank you to Dr. Benjamin Webb for your great mentorship, patience, and encouragement. I would not be here without your guidance.

# Contents

# List of Figures

# List of Tables

# Chapter 1. Introduction

The main objective of network science is to describe the structure of real-world networks and how that structure can lead to predictive models of the networks' behavior. One of the original surprises in this area was that many different kinds of networks, including various social, biological, and technological networks, had similar features such as degree distributions, the small-world property, hierarchical and community structure, etc. This suggests that there are similar mechanisms at work in the formation of diverse networks, or at least that these mechanisms result in similar statistical network features.

Although real-world networks often share similar features, their *topologies* (i.e., a network's underlying graph structure) are not identical. How the structure of real-world networks differs is complicated by both the size and the complexity of these networks. Most measures derived to distinguish one network from another need to scale well to handle the sheer size of these networks. Complexity, or the irregular structure of network links, similarly complicates comparing network structure as even similar networks (e.g., two social networks) can have widely varying structural features at many different scales.

In this thesis, we consider the use of persistent homology to compare the structure of various human family networks. Simply put, persistent homology is a method of representing holes or gaps in the structure of a network. This method is particularly useful as a tool for describing the structure of a network as it provides a way to measure how significant any particular hole is to the overall network structure. The basic idea involves "filling in" the network with simplices (points, edges, triangles, tetrahedra, etc.) and keeping track of how the network changes as we do so (see Chapter 3 for details).

The advantage of persistent homology is that it allows one to compare the holes across two networks without requiring a correspondence between the individual vertices or edges of the networks. This effectively deals with the two issues of *size* and *complexity* when comparing networks. This has also led to applications in a wide variety of fields such as geology [1], neuroscience [2], and even music theory [3]. These particular analyses reveal another benefit of using persistent homology: in each case the structural holes identified by persistent homology can be pulled back to recognizable features of the underlying networks. For example, Robins et al. have shown that the holes found using persistent homology correspond to percolating spheres through porous material [1]. In [2], structural holes arise when several groups of neurons are strongly connected sequentially, but out-of-sequence pairs are only weakly connected. Persistent homology provides a way to identify and classify these different sequences as well as quantify the strength of these connections. Structural holes or gaps can also represent abstract concepts, such as in [3] where holes are shown to correspond to the atonality in music compositions.

Persistent homology has been used to distinguish the structure of real-world networks including real and theoretical networks. In [16], Carstens and Horadam studied collaboration networks and showed that persistent homology could distinguish such networks from similar but randomly generated networks. Likewise, persistent homology is shown to distinguish model networks, biological networks, and user-interaction networks from each other in [15].

Here, we consider the use of persistent homology in distinguishing the structure of real-world networks, specifically human family networks. A lot of study has been done on the structure of social networks, but the structure of family networks is much less studied. Recent studies consider methods for constructing family networks from documents [7, 8], mating patterns [9], kinship models [10, 11], recent common ancestors [12, 13], and structured population modeling [14].

Using a proposed model to generate family networks, a specific goal of this thesis is to determine the extent to which persistent homology allows us to identify and compare real and modeled family networks (see Chapter 7). Holes or gaps found in family networks are likely quite different than those found in other networks such as social networks. Due to cultural, genetic, and potentially other reasons, unions in family networks typically form at specific distances. That is, distances between individuals who form unions, that may or may not result in children, are typically not too small or too large (see Chapter 2). This differs from social networks where social connections form at nearly any distance. Often these distances are very small as any two friends of the same individual may also become friends. This *triadic closure* is the formation of a friendship at a distance of 2 and is one of the major mechanisms of network growth in social networks, but not in family networks.

Network growth causes differences in network topology, which can be detected using persistent homology. Here we introduce a new method for representing persistent homology, which we call a persistence curve, and use this to study these structural similarities and differences (see Chapter 7). What we find is that the persistence curves of real family networks are similar to the persistence curves of modeled family networks. In fact, the persistence curves of subsets of real family networks, i.e., *sampled* family networks, are also similar to the persistence curves of modeled family networks. This suggests that even with incomplete data, family networks can be identified by their persistence curves. We introduce another tool for comparing persistent homology, the bottleneck distance. This is also capable of distinguishing these family networks, though it is more difficult to compute. An open area of research is understanding whether persistent homology can be used to distinguish other types of networks, e.g., social and biological networks, technological and information networks, or family and social networks, etc.

This thesis is organized as follows: In Chapter 2, we describe family networks and how they differ from social networks. In Chapter 3, we define the persistent homology of a network and introduce the notion of persistence curves. In Chapter 4, we define the bottleneck distance and show how both this distance and persistence curves can be used to compare networks. In Chapter 5, we describe the model we propose to generate family networks. In Chapter 6, we describe the family network data sets we use in our study and give our results in Chapter 7. We conclude by summarizing our results and future directions of this research.

# Chapter 2. Background

The *topology* of a network, which is the network's structure of connections, is typically represented by a graph. A *graph* $G = (V, E)$ is composed of a finite *vertex set* $V$ and *edge set* $E$. The vertex set $V$ represents the *elements* of the network, while the edges $E$ represent the links, or *relationships*, between these network elements. In social networks, such as Facebook and Twitter, the network elements are individuals and the edges represent friends and followers, respectively. In information networks, such as the World Wide Web, the network elements are web pages and the edges are the hyperlinks between them. Of particular interest in this thesis are family networks, in which elements represent individual people and edges represent familial relationships.

If a graph $G = (V, E)$ represents a family network, we let $V = \{1, 2, \ldots, n\}$ be the individuals within the network. The edges $E$ in a family network are of two types: *parent-child* edges $E_{PC}$ and *union* edges $E_U$, where a single edge is either a parent-child edge or a union edge, but not both. For the sake of simplicity, the edges $E = E_{PC} \cup E_U$ are considered to be undirected. That is, each edge indicates an undirected relationship either between parent and child or between couples. We use the comprehensive term *couple* to include all partners, marriages, cohabitants, etc. that may or may not have children. In the family networks we consider, union edges exist between all *couples*, and we will refer to such couples as *unions*.

The structure of a family network is often thought of or referred to as being "tree-like." Formally, a *tree* is a connected graph that does not have any cycles. A *cycle* refers to a sequence of vertices $1, 2, \ldots, m$ such that there is an edge $\{i, i+1\} \in E$ between individuals $i$ and $i+1$ for $i = 1, 2, \ldots, m-1$ and $m = 1$. This idea that a family is tree-like presumably comes from the fact that a family network is often constructed from an individual, their parents, their grandparents, and so on, ignoring all other edges. The result is a *family tree*. However, a common example of a cycle in a family network is the triangle consisting of two

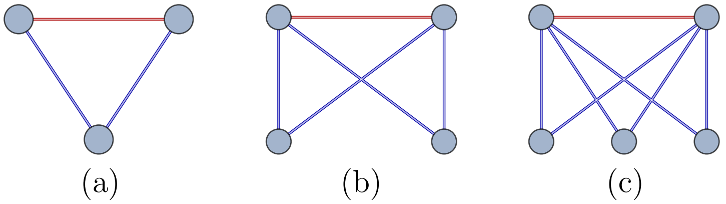
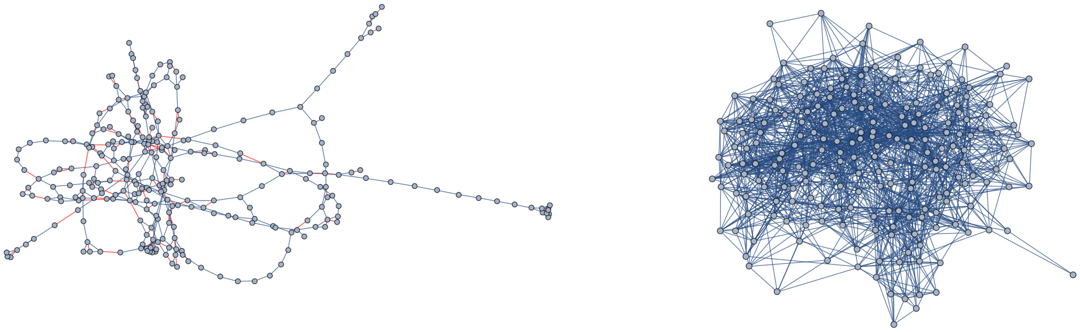


Figure 2.1: Familial cycles are cycles found within a union and their children. Union edges are shown in red and parent-child edges are shown in blue. Every child and their two parents form a trivial cycle as seen in (a). Two children and two parents form trivial cycles and a cycle of length 4 as seen in (b). Families with more than two children form only trivial cycles or cycles of length 4 as seen in (c).

parents and a child, with the two parent-child edges and one union edge that connects them (see Figure 2.1a). Because of the frequency of such cycles and the fact that they are the smallest possible cycles in a family network, we refer to them as *trivial cycles*. The only other possible *familial cycle*, or cycle found within a union and their children, is a cycle of length four, consisting of two parents and two children (see Figure 2.1b).

Although familial cycles are ubiquitous in family networks, they are not the only cycles that can form. Going far enough through an individual's ancestors, it is often possible to find a *nearest common ancestor*, i.e., a common ancestor of one's father and mother. If such an ancestor exists, then the family network has a nontrivial cycle. We refer to this as a *common ancestor cycle*, which consists of only parent-child edges. Other nontrivial cycles are possible in family networks via unions. For example, double cousins, which occur when two siblings from one family form unions with two siblings from another family. The result is a *mixed cycle*, or a cycle that contains both union and parent-child edges. In family networks, union and parent-child edges can combine in any number of ways to create complex nontree-like structures (see Figure 2.2 left).

A feature that is potentially unique to family networks is that union edges typically form at specific distances within these networks. Here the distance $d(i, j)$ between $i$ and $j$ is the shortest distance between these individuals if such a path exists, otherwise it is infinite. In the family networks we study (see Chapter 6), if a union edge $\{i, j\}$ forms between a couple,



Tikopia Family Network

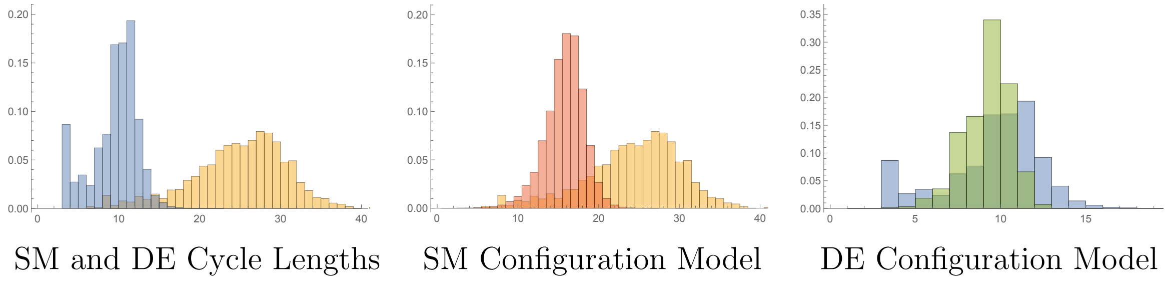
Residence Hall Social Network

Figure 2.2: Left: The largest connected component of the Tikopia family network consisting of 288 individuals is shown [4]. Parent-child edges are shown in blue and union edges are shown in red. Right: The largest connected component of the Residence Hall social network consisting of 217 individuals is shown [5].

the distance between individuals $i$ and $j$ prior to this formation is normally not very small, relatively speaking. Although there are cultural, genetic, and other reasons for this, one of the natural consequences of this behavior is that family networks do not typically have small, nonfamilial cycles, but potentially very large, extended cycles.

This phenomena can be seen in Figure 2.3. Shown left in orange is the probability distribution of the nonfamilial cycle lengths of the San Marino family network, a genealogical network of the population of the Republic of San Marino from the 15th to the end of the 19th century [4]. In this network, which consists of $28,586$ individuals, there are $7,146$ familial cycles of length three and $8,636$ familial cycles of length four. These are omitted in the figure so we can observe the nonfamilial cycle length distribution. In blue, is the probability distribution of all cycle lengths found in the Deezer Europe social network consisting of $28,281$ individuals [6]. Deezer is an online music streaming platform whose social network represents users in Europe and mutual user-follower relationships. Noticeably, the San Marino network has relatively few nonfamilial cycles under length ten but quite a few cycles with lengths greater than thirty. In contrast, the Deezer social network has a much tighter distribution of cycles ranging from roughly five to fifteen in length.

To understand the extent to which these distributions are related to the local structure of



SM and DE Cycle Lengths SM Configuration Model DE Configuration Model

Figure 2.3: Left: The probability distribution of the nonfamilial cycle lengths in the San Marino (SM) family network are shown in orange. The probability distribution of all cycle lengths is shown in blue for the Deezer Europe (DE) social network. Center: Shown in orange is again the cycle length distribution of the San Marino family network. In red is the probability distribution of the cycle lengths averaged over ten realizations of the configuration model on the San Marino network. Right: Shown in blue is again the cycle length distribution of the Deezer social network. In green is the probability distribution of the cycle lengths averaged over ten realizations of the configuration model on the Deezer social network.

the associated networks, we compare this to the cycle distribution of the associated configuration models. The *configuration model* is a method for generating random networks from a given degree sequence. Taking the degree sequences from both the San Marino family and Deezer social networks, we create ten versions of these networks each with the same degree sequences. The result of averaging the cycle length distributions of these versions of the San Marino and Deezer networks is shown in Figure 2.3 (center and right in red and green, respectively). While the cycle distribution for the San Marino network is quite different than what the configuration model predicts, the Deezer social network is quite similar to the distribution predicted by its configuration model. This suggests that much of the cycle structure in the Deezer social network is dominated by local interactions, whereas the cycles in the San Marino family network are affected by nonlocal mechanisms. This includes, presumably, the nonlocal *distance to union* phenomena described above.

Examining the cycle distributions emphasizes the role of cycles in distinguishing family networks from social or other real-world networks. However, the distribution alone fails to capture information on how the cycles interact or relate to each other. Tools from persistent homology allow us to identify cycles that fundamentally represent the entire distribution of cycles. In turn, this provides a way to describe and measure the relationship between any two cycles. This allows us to obtain the cycle structure of the network as a rich mathematical object, in place of the simple enumeration of the cycle distribution.

# Chapter 3. Persistent Homology of Networks

Persistent homology provides a method for studying cycles in a network. For the purposes of this thesis, a brief explanation will be given here. We start with a network $G = (V, E)$ and the distance $d(i, j)$ as defined in Chapter 2. We then create the distance matrix $D(G) = [d_{ij}]$ where the entry $d_{ij} = d(i, j)$ is the length of the shortest path between individuals $i$ and $j$. For each distance value $\delta$ that appears in the distance matrix $D(G)$, we form a simplicial complex $G_\delta$ as follows:

The set of 0-simplices is the set of vertices of $G$. The set of 1-simplices $E_\delta$ is the set of edges $\{i, j\}$ such that $d(i, j) \leq \delta$. Likewise, the set of $n$-simplices consists of $n$-simplices $[a_0 \quad a_1 \quad \ldots \quad a_n]$ such that $d(a_i, a_j) \leq \delta$ for $0 \leq i < j \leq n$. We note the following important properties of this construction. First, for $\delta < \epsilon$, $G_\delta$ is a subcomplex of $G_\epsilon$. Next, for $\delta \geq 1$, $G$ can be identified with a subcomplex of $G_\delta$. Finally, we assume $G = (V, E)$ is finite and let $M$ be the maximum value of $d(i, j)$, then for all $\delta \geq M$, $G_\delta = G_M$. We extend this definition of $G_\delta$ for all $i \in \mathbb{Z}, i \geq 0$, as follows: Given $i$, let $\delta_i$ be the greatest entry of $D(G)$ such that $\delta_i \leq i$. Then let $G_i = G_{\delta_i}$.

**Example 3.1. (Hexagonal Network)** Consider the *hexagonal network* $G = (V, E)$ with six vertices, forming a single cycle, shown in Figure 3.1b. This network has the distance matrix

$$D(G) = \begin{bmatrix} 0 & 1 & 2 & 3 & 2 & 1 \\ 1 & 0 & 1 & 2 & 3 & 2 \\ 2 & 1 & 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 & 1 & 2 \\ 2 & 3 & 2 & 1 & 0 & 1 \\ 1 & 2 & 3 & 2 & 1 & 0 \end{bmatrix}.$$

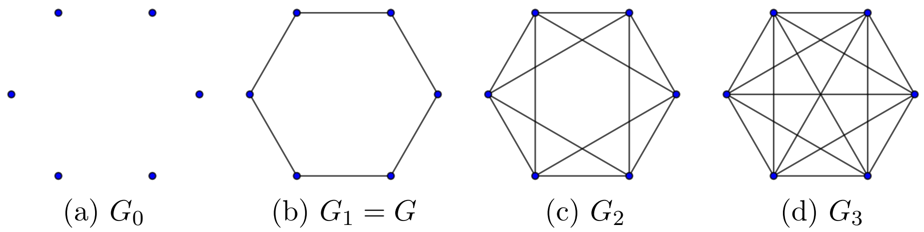


Figure 3.1: The hexagonal network from Example 3.1 is filled in as $i$ varies from 0 to 3. This produces the graphs $G_0, G_1, G_2, G_3$ shown left to right.

For the values $i = 0$, 1, 2, 3, we form four simplicial complexes, $G_0$, $G_1$, $G_2$, and $G_3$ as shown in Figure 3.1. For $i = 0$, $E_0$ is empty. Furthermore, since there are no edges, there can be no higher dimensional simplices either. Thus, $G_0$ consists of six vertices. For $i = 1$, $E_1$ contains the six edges that form the network's single cycle, so $G_1$ has the same vertices and edges as $G$. This graph has no trivial cycles, so $G_1$ contains no simplices of dimension greater than 1. For $i = 2$, $E_2$ gains six additional edges. We also now have eight trivial cycles. Each of these is the boundary of a 2-simplex, so $G_2$ contains these eight 2-simplices as well. However, no subset of these 2-simplices forms the boundary of a 3-simplex, so $G_2$ has no simplices of dimension greater than 2. For $i = 3$, $E_3$ contains all possible edges between the vertices of $G$, so all possible trivial cycles are present. Additionally, all possible 2-simplices, and hence all possible $n$-simplices, are also present in $G_3$. In particular, $G_3$ is a 6-simplex with its boundary. Since 3 is the largest value we see in the distance matrix, then $G_i = G_3$ for $i \in \mathbb{Z}, i > 3$.

The persistent homology of the network $G$ measures how the homology of $G_i$ changes as $i$ increases. If certain features can be identified across multiple values of $i$, we say they *persist*. Intuitively, features that arise from the actual network structure should persist for many values of $i$, while features that arise because of measurement error, or "noise", should only appear sporadically. Here we use $H_p(G_i)$ to denote the dimension-$p$ homology of $G_i$ with coefficients in $\mathbb{Z}_2$ and note that $H_p(X)$ is a vector space of $\mathbb{Z}_2$. We likewise use $H(X)$ to denote the total homology of $X$ with coefficients in $\mathbb{Z}_2$. This allows us to give the following definition:

**Definition 3.2. (pth Persistent Homology)** For a network $G$, and integers $i, j$ with $0 \leq i \leq j$, let the function $\phi_{i,j} : H_p(G_i) \to H_p(G_j)$ be the linear map induced by the inclusion $G_i \to G_j$. The $p$th persistent homology of $G$, $PH_p(G)$ is the pair $(\{H_p(G_i)\}_{i \geq 0}, \{\phi_{i,j}\}_{0 \leq i < j})$.

Our analysis in Chapters 4 and 7 only requires the first few dimensions of persistent homology to distinguish the networks we consider. As such, we will provide equivalent definitions for $PH_0$, $PH_1$, and $PH_2$ using network concepts. We also illustrate these definitions on the hexagonal network in Example 3.1. While the equivalent definitions will rely on a choice of bases for $H_p(G_i)$, the Fundamental Theorem of Persistent Homology [17] ensures that bases can be chosen for each $H_p(G_i)$ in a compatible fashion. That is, we can ensure that the basis elements of $H_p(G_i)$ are mapped forward to either basis elements or 0; and that no two basis elements are sent to the same basis element. There may be many such compatible bases, but our analysis is independent of the choice of compatible bases.

**Definition 3.3. (Births and Deaths)** Let $G = (V, E)$ be a network with corresponding simplicial complexes $G_0, G_1, G_2, \cdots$. The $p$th persistent homology of $G$ provides maps $\phi_{i,j}$ between the $p$th homology of $G_i$ and the $p$th homology of $G_j$. Suppose that basis elements have been chosen for each $H_p(G_i)$ so that if $\alpha$ is a basis element of $H_p(G_i)$, $\phi_{i,j}(\alpha)$ is either trivial in $H_p(G_j)$ or a basis element of $H_p(G_j)$. The birth of a basis element $\alpha \in H_p(G_j)$ is the minimum index $i$ such that $\alpha = \phi_{i,j}(\hat{\alpha})$ for some basis element $\hat{\alpha} \in G_i$. The death of $\alpha$ is the minimum index $k$ such that $\phi_{j,k}(\alpha)$ is trivial.

To improve our intuition, instead of considering $\alpha \in H_p(G_j)$, we can choose a representative object in $G_j$. We will demonstrate how to choose such representatives for $H_0$, $H_1$, and $H_2$ in the following definitions. Given such representatives, though, the maps $\phi_{i,j}$ and $\phi_{j,k}$ correspond to the inclusion maps $G_i \subset G_j \subset G_k$. Choosing a compatible basis ensures that we can choose a single representative object that corresponds to $\alpha \in H_p(G_j)$, $\hat{\alpha} \in H_p(G_i)$, and $\phi_{j,k}(\alpha) \in H_p(G_k)$. The birth of $\alpha$ is then just the first $G_i$ in which the representative exists, and the death of $\alpha$ is the first $G_k$ in which the representative is *null-homotopic*. The representative is null-homotopic if it is the sum of trivial point pairs in $H_0$, the sum of trivial cycles in $H_1$, or the sum of trivial surfaces in $H_2$.

**Definition 3.4. (Representing Persistent Homology: Dimension 0)** Let $G = (V, E)$ be a network with $n$ distinct vertices, $V = \{1, 2, \dots, n\}$, forming $k$ connected components. $H_0(G_0) \cong \mathbb{Z}_2^n$, so we may identify the basis for $H_0(G_0)$ with the set of all $n$ vertices. Likewise, we may choose $k$ vertices, one from each connected component, to represent the basis for $H_0(G_i) \cong \mathbb{Z}_2^k$ for $i \geq 1$. Thus, we will refer to the vertices of $G$ as *elements* of $PH_0(G)$.

**Example 3.5.** We now consider $PH_0(G)$ for the hexagonal network $G$ in Example 3.1, with $G_0$, $G_1$, $G_2$, and $G_3$ as given in that example. Recall that $G$ has six distinct vertices forming one connected component. If we take any numbering of the vertices, $V = \{1, 2, 3, 4, 5, 6\}$, then $H_0(G_0) \cong \mathbb{Z}_2^6$, which is equivalent to the vector space over $\mathbb{Z}_2$ with basis $V$. For $i > 0$, $H_0(G_i) \cong \mathbb{Z}_2$, which is equivalent to the vector space over $\mathbb{Z}_2$ with basis $\{1\}$. For any $v \in V$, and since $i = 0$ is the first time we see $v$, we call this the *birth* of $v$. At $i = 1$, and since we have removed all vertices except 1 from the basis, we say this is the *death* of those five 0-simplices (see Figure 3.1a-d). Since 1 will always be in the basis for $G_i$, the *death* of 1 is said to be $\infty$.

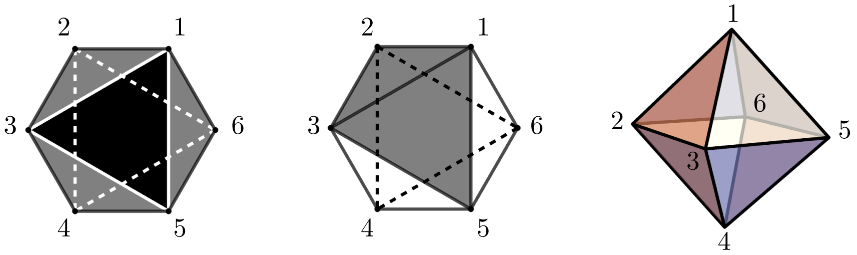
**Definition 3.6. (Representing Persistent Homology: Dimension 1)** Let $G = (V, E)$ be a network with one connected component. For each $i \geq 0$, we can identify the basis of $H_1(G_i)$ with a set $C_i$ of cycles in $G_i$. The Fundamental Theorem of Persistent Homology allows us to choose these cycles so that if $\sigma$ is a cycle in $C_i$, then exactly one of the following is true for any integer $j \geq 0$:

(i) $\sigma$ does not exist in $G_j$, in which case $j < i$;

(ii) $\sigma$ is trivial or null-homotopic in $G_j$, in which case $i < j$;

(iii) $\sigma$ is a cycle in $C_j$.

Thus, we will refer to the cycles in $\bigcup_{i \geq 0} C_i$ as the *elements* of $PH_1(G)$.



(a) $G_2$ trivial cycles (b) $G_2$ null-homotopic cycle (c) $G_2$ sphere

Figure 3.2: A visual depiction of 2-simplices and null-homotopic cycles in $G_2$ for the hexagonal network. Left: Four 2-simplices: $[1 \quad 2 \quad 3]$, $[3 \quad 4 \quad 5]$, $[1 \quad 5 \quad 6]$, and $[1 \quad 3 \quad 5]$. Center: A non-trivial, but null-homotopic cycle, $1, 2, 3, 5, 1$ filled in by two 2-simplices $[1 \quad 2 \quad 3]$ and $[1 \quad 3 \quad 5]$. Right: All eight 2-simplices represented as the faces of a regular octahedron.

We note that $C_0$ is always empty, since there are no edges in $G_0$, and $\text{rank}(H_1(G_i)) = |C_i|$ for all $i \geq 0$. Because of the construction of $G_i$, all elements of $PH_1(G)$ will be present in $G_1$. One can think of the elements of $PH_1(G)$ as representing "large" cycles. More specifically, if a cycle $\sigma$ is contained in $\bigcap_{s \leq i \leq t} C_i$, then it must have a diameter of at least $t$ and at least one pair of consecutive vertices distance $s$ apart.

**Example 3.7.** We now consider $PH_1(G)$ for the hexagonal network $G$ in Example 3.1. In Figure 3.1a and 3.1b we see that $G_0$ has no cycles, $G_1$ has exactly one cycle, and the cycle in $G_1$ is non-trivial. In Figures 3.2a and 3.2b, we have indicated some of the cycles in $G_2$, namely the cycles 1,2,3,1; 3,4,5,3; 1,5,6,1; and 1,3,5,1 in Figure 3.2a and the cycles 1,2,3,5,1 in Figure 3.2b. In fact, Figure 3.2c shows us that $G_2$ is an octahedron and therefore every cycle in $G_2$ is either trivial or null-homotopic. Finally, $G_3$ contains even more cycles than $G_2$, such as 1,3,6,1; but these are all null-homotopic since $G_3$ also contains every possible 2-simplex for six nodes. Therefore, $PH_1(G)$ consists of only one cycle, 1,2,3,4,5,6,1; which appears in $G_1$, so we say that $t = 1$ is the *birth* of the cycle. The cycle is null-homotopic in $G_2$, so $t = 2$ is the *death* of the cycle.

We now turn our attention to $PH_2(G)$, but in order to represent $PH_2(G)$ we need to introduce some new structure for the induced networks. A *triangle* $[a \quad b \quad c]$ in $G_i$ is a set of three vertices, $a$, $b$, and $c$, that form a trivial cycle in $G_i$. That is, the edges $\{a, b\}$, $\{b, c\}$, and $\{a, c\}$ are all present in $G_i$. A *closed surface* in $G_i$ is a set of distinct triangles so that for each $[a \quad b \quad c]$ in the set there is exactly one other triangle $[a \quad b \quad d]$ also in the set. A closed surface in $G_i$ is *trivial* if the corresponding set of 2-simplices is null-homotopic in $G_i$. That is, the closed surface is "filled in" by some collection of 3-simplices in $G_i$. For example, the octahedron in Figure 3.2c is a non-trivial closed surface. However, if we were to add the edge $\{1, 4\}$ and all corresponding 3-simplices, the octahedron would be filled in by four tetrahedra: $[1 \quad 3 \quad 4 \quad 5]$, $[1 \quad 2 \quad 4 \quad 3]$, $[1 \quad 6 \quad 4 \quad 2]$, and $[1 \quad 5 \quad 4 \quad 6]$.

**Definition 3.8. (Representing Persistent Homology: Dimension 2)** Let $G = (V, E)$ be a network with one connected component. For each $i \geq 0$, we can identify the basis for $H_2(G_i)$ with a set $S_i$ of non-trivial closed surfaces in $G_i$. The Fundamental Theorem of Persistent Homology allows us to choose these representatives so that if $\sigma$ is a closed surface in $S_i$, then exactly one of the following is true for any integer $j \geq 0$

(i) $\sigma$ does not exist in $G_j$, in which case $j < i$,

(ii) $\sigma$ is trivial in $G_j$, in which case $i < j$,

(iii) $\sigma$ is a cycle in $S_j$.

Thus we will refer to the closed surfaces in $\bigcup_{i \geq 0} S_i$ as the *elements* of $PH_2(G)$.

The geometric intuition for $PH_2(G)$ is similar to that of $PH_1(G)$ in identifying large 'voids' in $G$. If $\sigma \in \bigcap_{s \leq i \leq t} S_i$, then $\sigma$ is a closed surface with diameter at least $t$. The value of $s$ is harder to describe, but is related to the density of vertices.

**Example 3.9.** We now consider $PH_2(G)$ for the hexagonal graph $G$ in Example 3.1. Recall from Example 3.7 that $G_0$ and $G_1$ have no trivial cycles, and therefore contain no closed surfaces. We can see in Figure 3.2 that $G_2$ has exactly one closed surface and it must be non-trivial, since there are no 3-simplices. Finally, $G_3$ has many closed surfaces, but because it contains every possible 3-simplex on six nodes, these are all trivial. Therefore, $PH_2(G)$ consists of only one closed surface, which appears in $G_2$, so we say $t = 2$ is the *birth* of the surface. The closed surface is trivial in $G_3$, so this is the *death* of the surface.

**Definition 3.10. (Persistence Intervals)** Given an *element $\sigma$ (vertex, cycle, or closed surface) of the persistent homology* of a network $G$, the *birth* of $\sigma$ is the smallest integer $i$ so that $\sigma \in G_i$. The *death* of $\sigma$ is the largest integer $j$ so that $\sigma \in G_{j-1}$ and $\sigma$ is trivial in $G_k$ for $k \geq j$, if such an integer exists. Otherwise, the death of $\sigma$ is said to be $\infty$. The *persistence interval* for $\sigma$ with birth $a$ and death $b$, is $[a, b)$. This represents the set of all parameter values $i$ for which the equivalence class of $\sigma$ is a non-trivial element of $H(G_i)$. The *persistence* of $\sigma$ in this case is $b - a$.

**Example 3.11.** We now finish our consideration of the persistent homology of $G$ from Example 3.1. Recall from Example 3.5 that $PH_0(G)$ has six elements. These all have birth $t = 0$. Five of these have a death of $t = 1$, and one of these has a death of $\infty$. Therefore, the persistence intervals for $PH_0(G)$ are $[0, 1) \times 5$, $[0, \infty)$. From Example 3.7, we know $PH_1(G)$ has one element, with birth $t = 1$ and death $t = 2$. Therefore, the persistence interval for that element is $[1, 2)$. Note that the diameter of the cycle is 3 and every pair of consecutive vertices is distance 1 apart. From Example 3.9, $PH_2(G)$ has one element, with birth $t = 2$ and death $t = 3$. Therefore the persistence interval for that element is $[2, 3)$. Note that the diameter of the corresponding set of vertices is 3 in $G$.

Given the representatives chosen in Definitions 3.4, 3.6, 3.8, and 3.10, we have the following three observations regarding the persistent homology of a finite, undirected, unweighted graph $G$:

(i) If $G$ has $n$ nodes, then $PH_0(G)$ will have exactly $n$ intervals, with exactly one $[0, \infty)$ interval for each connected component and the rest will be $[0, 1)$ intervals.

(ii) In dimension 1, $PH_1(G)$ describes the number and sizes of the non-trivial cycles in the original network. The intervals will all be of the form $[1, b)$ for some integer $b > 1$. The value of $b$ is related to the diameter of the corresponding cycle. In the networks we have studied, we note that a persistence interval $[1, b)$ in $PH_1(G)$ corresponds to a simple cycle with between $3b - 2$ and $3b$ nodes, inclusive.

(iii) In dimension 2, the voids we detect in $PH_2(G)$ indicate non-trivial intersections of cycles. Though we do not have a precise formula, we can say that when large cycles intersect non-trivially we obtain voids with high persistence values.

# Chapter 4. Comparing Networks

In this chapter, we demonstrate how methods based on persistent homology can be used to compare different networks. The two methods we introduce in this thesis are based on using (a) the bottleneck distance and (b) persistence curves of a given set of networks. Both (a) and (b) rely on first computing persistence intervals and then analyzing the differences in these intervals.

The two networks we consider throughout this section to demonstrate these methods are the Tikopia family network from Figure 2.2 (left) and the hexagonal network from Figure 3.1. The persistence intervals for these networks is given in Table 4.1, respectively.

## 4.1 Persistence Diagrams and Bottleneck Distance

One common way to represent persistence intervals is to plot them as points in $\mathbb{R}\times(\mathbb{R}\cup\{\infty\})$, which is usually called a persistence diagram. While this does not indicate how often a given persistence interval occurs, it does provide information on what kind of persistence intervals occur for a given network.

| Dimension | Interval Type and Persistence | |
|---|---|---|
| | Tikopia | Hexagon |
| Dimension 0 | $[0,\infty)\times 8,\ [0,1)\times 286$ | $[0,\infty)\times 1,\ [0,1)\times 1$ |
| Dimension 1 | $[1,2)\times 16$, $[1,3)\times 19$, $[1,4)\times 5$,<br>$[1,5)\times 3$, $[1,6)\times 2$, $[1,7)\times 1$ | $[1,2)\times 1$ |
| Dimension 2 | $[2,3)\times 4$, $[3,4)\times 11$, $[4,5)\times 12$,<br>$[5,6)\times 4$, $[6,7)\times 5$, $[7,8)\times 1$, $[8,9)\times 1$ | $[2,3)\times 1$ |

Table 4.1: The persistence intervals of the Tikopia family network and the hexagon network are shown. Here the notation $[a,b)\times k$ indicates that the network has $k$ persistence intervals $[a,b)$. The corresponding persistence diagrams are shown in Figure 4.1 and the corresponding persistence curve for the Tikopia network is shown in Figure 4.2.

**Definition 4.1.** Let $PH_p(G)$ be the $p$th persistent homology of a network $G$. The persistence diagram for $PH_p(G)$ is a multiset of points in $\mathbb{R} \times (\mathbb{R} \cup \{\infty\})$ defined as follows:

- For each $\sigma \in PH_p(G)$ with persistence interval $[a, b)$, we include one copy of the point $(a, b)$.
- For each $c \in \mathbb{R}$, we include infinitely many copies of the point $(c, c)$.

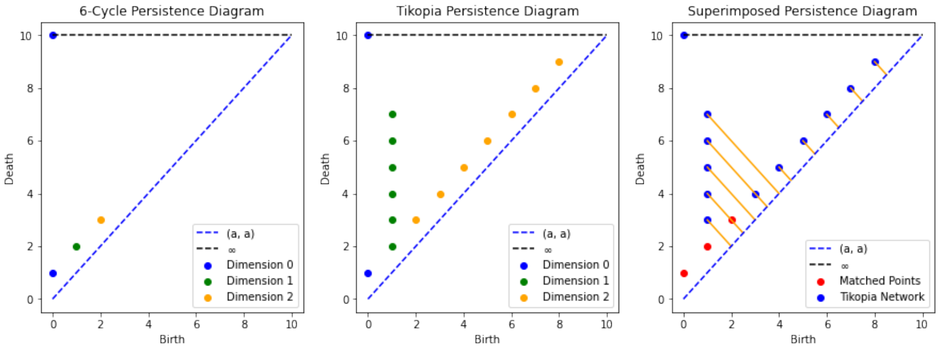
Note that we include the points $(a, a)$ to represent features in $G$ that are considered trivial in $PH_p(G)$, such as cycles consisting of exactly three vertices. This inclusion is necessary for us to define a meaningful metric on the space of persistence diagrams. The metric we use here is called the bottleneck distance.

**Definition 4.2. (Bottleneck Distance)** Let $S_1$ and $S_2$ be persistence diagrams for two networks $G$ and $H$, respectively. Let $\eta$ range over the set of bijections from $S_1$ to $S_2$, then the *bottleneck distance* between $S_1$ and $S_2$ is

$$d_B(S_1, S_2) = \inf_{\eta} \sup_{x \in S_1} \|x - \eta(x)\|_\infty.$$

The Fundamental Theorem of Persistent Homology [17] ensures that if two graphs are isomorphic, the corresponding persistence diagrams will be equal, and thus the bottleneck distance will be 0. However, it is possible for non-isomorphic graphs to have identical persistence diagrams. For example, let $G$ be the graph given by $V = \mathbb{Z}$, $E = \{\{a, a+1\}, a \in \mathbb{Z}\}$, and let $H$ be the graph given by $V = \mathbb{N}$, $E = \{\{a, a+1\}, a \in \mathbb{N}\}$. These graphs are non-isomorphic, but the persistence diagram for either graph will consist of one point of the form $(0, \infty)$, and infinitely many copies of the points $(0, 1)$ and $(c, c)$ for $c \in \mathbb{R}$.

**Example 4.3.** Notice that the persistence intervals for the Tikopia family network (see Table 4.1) include, as a subset, the persistence intervals from the hexagonal network we considered in Example 3.11. We can form a bijection between the persistence diagrams of the Tikopia and hexagonal network by identifying the non-trivial intervals from the hexagonal network



Hexagonal Network PD Tikopia Network PD Bottleneck Bijection

Figure 4.1: Left: The persistence diagram (PD) of the hexagonal network in Figure 3.1 is shown. Center: The persistence diagram of the Tikopia Family network in Figure 2.2 (left) is shown. Right: A bottleneck bijection between the persistence intervals of the hexagonal and Tikopia family network is shown. Orange lines show which points are matched to points of the form $(a, a)$ where $a \in \mathbb{R}$.

with those of the Tikopia network. We then map any additional intervals from the Tikopia network of the form $[a, b)$ to the trivial interval $[\frac{a+b}{2}, \frac{a+b}{2})$.

This mapping is shown in Figure 4.1 (right). Here, $[1, 7)$ is mapped to $[4, 4)$. As this pair of points is farther apart than any other pair in this bijection, the bottleneck distance for the two networks is *at most three*, since we take an infimum over all possible bijections. Conversely, there is no interval in the hexagonal persistence diagram that is closer to $[1, 7)$ than 3, so the bottleneck distance is *at least three*. Therefore, we can conclude that the bottleneck distance for these two persistence diagrams is exactly 3.

Suppose that two networks, each of which is connected, admit isometric embeddings in $\mathbb{R}^n$. The Stability Theorem [18] guarantees that if the Hausdorff distance between the embeddings is $\delta$, then the bottleneck distance for the corresponding persistence diagrams is at most $\delta$. Therefore, if we find a significant difference between two networks using the bottleneck distance, that difference is significant in the underlying network structure. For example, if the $PH_1$ persistence diagrams differ by $\delta$, then any attempt to pair up cycles in the networks must include at least one pair of cycles for any isometric embedding that are $\delta$ apart in that embedding. In Chapter 7, we apply this idea to family networks.

## 4.2 Persistence Curves

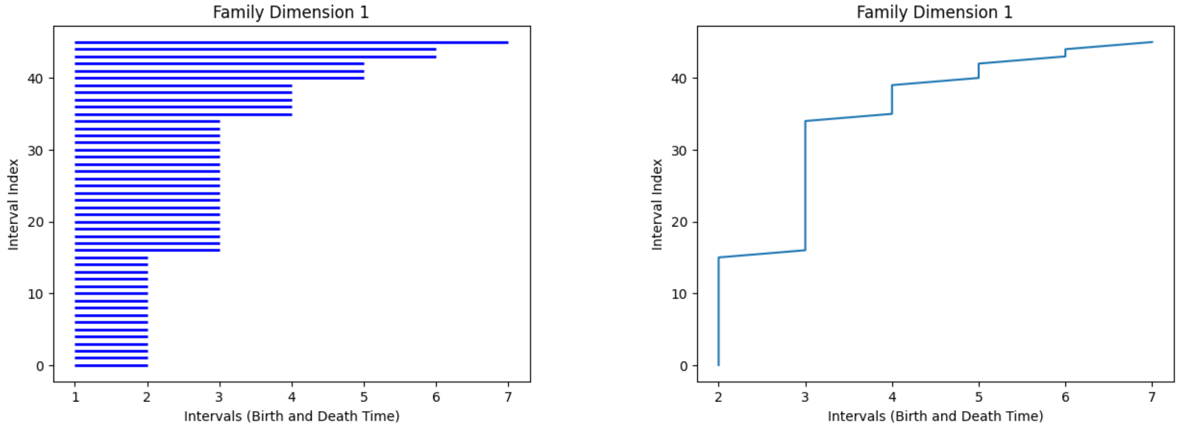
For the data we consider, persistence diagrams obfuscate a key difference that we deem important: the number of persistence intervals. For a simple example of this, consider networks of the form $V = \{1, 2, \ldots, n\}$ with edges of the form $\{i, i+1\}$ for $1 \leq i < n$. For $n \geq 2$, any network of this type will have persistence intervals $[0, 1) \times (n-1)$ and $[0, \infty) \times 1$. However, when plotting the persistence diagram we will only 'see' two points: $(0, 1)$ and $(0, \infty)$.

To address this limitation, we introduce the notion of a *persistence curve* as a new way to visualize the persistent homology of a network (see Definition 4.4). The difference between the persistence curve and the persistence diagram of a network is that the persistence curve also includes the number of intervals of a particular type. To create a persistence curve we first compute a network's persistence intervals, then sort the intervals of a given dimension by their persistence into a bar graph. For instance, in dimension-one the Tikopia family network has thirteen $[1, 2)$ intervals, nineteen $[1, 3)$ intervals, etc., which are sequentially stacked as shown in Figure 4.2 (left) to create what we will call a *barcode*. To create the associated persistence curve we connect the endpoints of each subsequent bar as shown in Figure 4.2 (right).

In dimension-one, the birth times of our intervals will all start at 1, as our networks are unweighted, undirected, and connected. This means that for this dimension the resulting bar graph is also a plot of the death times for each interval. For the other higher dimensions, which have varied birth times, we also plot the lengths of the intervals, but for simplicity we start at 1 as in dimension-one.

A formal definition of a network's persistence curves is the following:

**Definition 4.4. (Persistence Curves)** Let $G = (V, E)$ be a network with nonempty vertex and edge sets. Then let $\{[a_j, b_j)\}$ be the set of all persistence intervals for each $\sigma_j \in PH_n(G)$ where $j \in \mathbb{N}$. For all $n \in \mathbb{N}$, the persistence curve $PH_n(G)$ is the linear interpolation of the set of points $\{(b_j - (a_j - 1), j)\}$ where $b_{j-1} - (a_{j-1} - 1) \leq b_j - (a_j - 1)$.



Tikopia Network Barcode

Tikopia Network Persistence Curve

Figure 4.2: Left: The barcode of the Tikopia family network in dimension-one is shown. The individual bars are formed from the persistence intervals given in Table 4.1. Right: The associated persistence curve is shown.

Visualizing the data in this way allows us to compare the persistent homology of different networks in a similar fashion to persistence diagrams while retaining different information. In particular, we can see how many intervals there are of a given persistence, whereas the persistence diagram only indicates the presence of such an interval. We will typically plot persistence curves of multiple networks on the same axes to give a preliminary idea of what differences exist in the persistent homology of different networks (see Chapter 7).

# Chapter 5. Model of a Human Family Network

The main goal of this thesis is to create a model that can generate a realistic family network. Specifically, using data from a real-world family network, our goal is to create a family network with similar local and global features. The hope is that our model will give insight as to what characterizes family networks. We hypothesize that (i) the distance at which unions form and (ii) the number of children each union has are fundamental features that determine the structure of family networks. As such, the model we propose uses these two features to model the growth of a human family.

## 5.1 Model Parameters

Given a real-world family network $F = (V, E)$, our model seeks to create a family network $M_F = (V_F, E_F)$ that mimics the features of the real family network. The features we seek to recreate are the network's (i) distribution of union distances $U = U(F)$ and (ii) distribution of the number of children per union $C = C(F)$.

As defined in Chapter 2, union distances are found by computing the shortest path between two given vertices that form a union. This path is found by deleting the union edge between the two vertices and all of their parent-child edges, so that the shortest path is found outside the immediate family via other parent-child edges. The idea is that this path gives the shortest distance between the pair before their union is formed. If there is no path connecting the two vertices, the distance is defined to be infinite. The finite union distances are collected to form the distribution $U$, where $P(U = d)$ is the probability that a randomly selected union in $F$ will have a distance $d < \infty$. The number of infinite distances are also counted and converted into the parameter $d_\infty$, equal to the number of union distances in $F$ that are infinite. A finite union distance indicates that the couple has a common ancestor, whereas an infinite union distance indicates that the couple does not have a common ancestor in the family network.

The second distribution we determine is the network's distribution of children $C$, which is found by counting the number of children each union shares. Here, the distribution $C$ has the property that $P(C = k)$ is the probability that a randomly selected union in $F$ has $k$ children. Based on the data we consider, it is possible for only one person in a union to have children (e.g., second marriages where only one spouse previously had children), but we have chosen to focus on children that both people in a given union are linked to via parent-child edges. Although this leaves out some children in the real-world family networks we consider, the number of children that are not counted using this convention is typically very small, and this simplification has little effect on the networks our model creates.

In addition to these network parameters and distributions, two union probabilities are

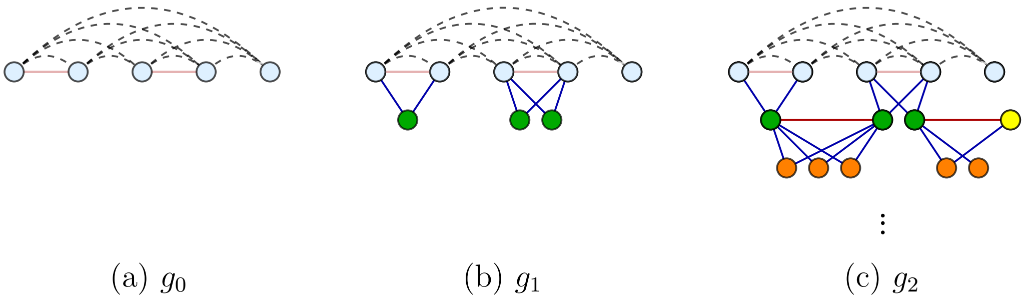

(a) $g_0$ (b) $g_1$ (c) $g_2$

Figure 5.1: An example of a growing family network is shown. The dashed, curved lines represent the initial distances of the initial set of $n_0$ vertices in $g_0$. Union edges are shown in red and parent-child edges are shown in blue. Left: The initial generation $g_0$ of the network with $n_0 = 5$ vertices is shown in light blue. Middle: The unions formed in $g_0$ have children (represented by green vertices) creating the next generation $g_1$. Right: A non-connected vertex (shown in yellow) is added to the network and unions are formed among $g_1$. These unions then have children (represented by orange vertices) to create the next generation $g_2$. The network continues to build generationally until all $g$ generations are formed.

computed using the real-world family network $F$. The first is the probability a vertex forms a union, $p_{union} \in [0, 1]$. This is found by taking the number of union edges multiplied by 2 and dividing by the total number of vertices in $V_F$. The number $p_{union}$ gives us the fraction of vertices in the network that form a union. The next probability we determine is the probability $p_\infty \in [0, 1]$ that a vertex is not initially connected to the network. These *non-connected* vertices are vertices that have formed a union with another vertex in the family network, but are not children of any other vertex in the network; their only initial link to the network is through their union edge. Thus, the number $p_\infty$ is the fraction of the vertices $V_F$ that are non-connected in $M_F$.

In our model, the growth of a family happens generationally. One can think of a family starting with a single couple or union that we'll call the initial generation $g_0$. This couple then has children to create a new generation $g_1$. Those children then form unions with others and have their own children to create the next generation $g_2$. This continues for several generations (Figure 5.1 gives an example of this process).

Our model replicates this growth on a larger scale by creating several unions in each

| **Model Parameters** | |
|---|---|
| $U$ | Distribution of finite union distances |
| $d_\infty$ | Number of infinite distances |
| $C$ | Distribution of number of children |
| $p_{union}$ | Probability of forming a union |
| $p_\infty$ | Probability of an infinite union |
| $n_0$ | Initial number of vertices |
| $g$ | Number of generations |

Table 5.1: Given a real-world family network $F = (V, E)$, the model creates the family network $M_F = (V_F, E_F)$ using the parameters shown.

generation. Thus, the last parameters needed for the model are the number of initial vertices $n_0 > 0$, and the generations $g > 0$ simulated in the network. These parameters determine the initial size and number of generations, respectively, in the resulting model of the family network $M_F = (V_F, E_F)$. (More details about how these two parameters are chosen is given in Chapter 6).

Table 5.1 summarizes the model parameters and distributions as described above.

## 5.2 Initialize Modeled Family Network

Given a family network $F = (V, E)$, the model begins with an empty graph $G = G(F)$ and adds $n_0$ initial vertices with no edges between these vertices. This set of vertices represents the initial generation $g_0$ of the modeled family network $M_F$. The $n_0 \times n_0$ distance matrix $D_0(G) = [d^0_{ij}]$ is then created. Unlike the distance matrix $D(G)$ that was defined in Chapter 3, $D_0(G)$ stores initial distances between each of the $n_0$ vertices. These distances are determined using the distribution $U = U(F)$. For each pair of vertices $i$ and $j$ in the initial generation, we choose the distance $d_{ij}$ using a list of all possible finite distances to randomly select a distance value $d > 0$ for each entry $d_{ij}$ in $D_0(G)$. Thus, the $ij$th entry of the resulting symmetric matrix represents the distance between vertices $i$ and $j$ in the initial generation. These distances are used later in the model to determine if a given pair of vertices will form a union, and each distance is updated as the network grows with each generation (see Section 5.3). That is, the distance matrix $D_\mu$ stores all previous distances and gets larger with each generation for $0 < \mu < g$.

The probability of each possible distance is found by first considering the distribution of finite union distances $U$. The probability mass function $P(U = d)$ gives the probability that a randomly selected union will have distance $d < \infty$. However, this distribution is restricted to the distances found in the original family network $F$. Thus, specific distance(s) $s$ could be missing from the data since no pair of vertices in $F$ form a union at that distance (see Figure 5.2a). This means our model would penalize that specific union distance $s$ from ever occurring. In our model, it is possible for vertices to be a distance $s$ apart, and using the distribution $P(U = d)$ would impede those vertices from ever forming a union together. As this is somewhat unrealistic, we will use a smoothing technique to incorporate any distance between the minimum and maximum distances found in $F$ by creating a probability density function (PDF) associated with the union distance distribution.

To calculate this PDF of the union distribution, we use a kernel density estimator (KDE). For $(x_1, x_2, ..., x_n)$ independent and identically distributed samples, which come from the union distribution $U$, the KDE is $\hat{f}(x; h) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x-x_i}{h})$, where $n$ is the number of samples, $h > 0$ is the bandwidth, and $K(x)$ is the kernel function [19]. Here, the bandwidth $h$ acts as a smoothing parameter. A larger bandwidth $h$ leads to a very smooth, or high-bias, density distribution. A smaller bandwidth $h$ leads to an unsmooth, or high-variance, density distribution [20]. In our model, we set $h = 1$ to balance the bias and variance of the distribution. In most of the data we consider, the finite distances are approximately normally distributed (see Figure 5.2a for an example), so a Gaussian kernel is used for the kernel density estimation. That is, $K(x) = \phi(x)$, where $\phi(x)$ is the standard normal density function.

After using the KDE to estimate the PDF, spline interpolation is used to fit the curve. We then find the probability of each distance $d$ by integrating around each distance using the interval $[d - 0.5, d + 0.5]$. For example, to find the probability of distance $d = 4$, we integrate the fitted curve over the interval 3.5 to 4.5. Doing this for each distance, we create the probability distribution $\hat{U} = \hat{U}(F)$, where $P(\hat{U} = d)$ is the probability of a finite distance $d$, given by

$$\int_{d-0.5}^{d+0.5} \hat{f}(x; h)dx,$$

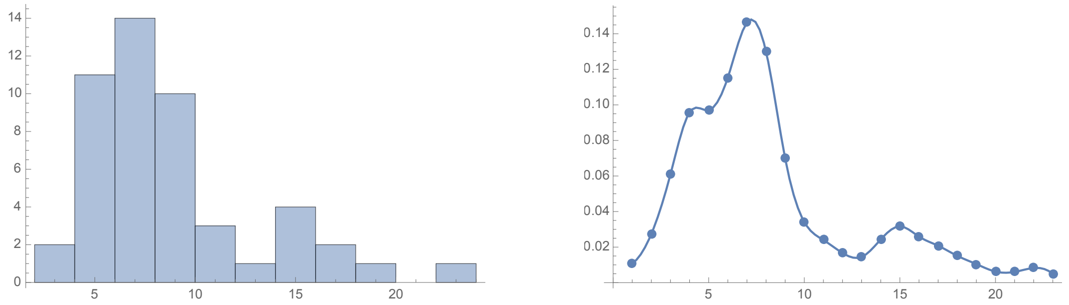
where $\min U \leq d \leq (\max U + 1)$ and $\min U$ is the smallest union distance in $F$ and $\max U$ is the largest union distance in $F$ (see Figure 5.2b). The probability of an infinite distance, $d_\infty$, is added to the distribution of finite distance probabilities $\hat{U}$, and all probabilities are normalized to create the distribution $\bar{U} = \bar{U}(F)$. That is, $P(\bar{U} = d)$ is the probability of the finite or infinite distance $d$, including any missing distances $s$, in the family network $F$. Thus, we have the probability

$$P(\bar{U} = d) = \lambda[P(\hat{U} = d) + d_\infty],$$

where $\lambda = \left[\left(\sum_{d=\min U}^{\max U} \int_{d-0.5}^{d+0.5} \hat{f}(x; h)dx\right) + d_\infty\right]^{-1}$ is a normalizing factor.

A similar process also produces a probability distribution of the number of children each union can have. That is, the PDF of the children distribution $C$ is estimated using the KDE $\hat{g}(x; h) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x-x_i}{h})$, where $(x_1, x_2, ..., x_n)$ are independent and identically distributed samples from $C$, $h = 1$, and $K(x) = \phi(x)$. Using spline interpolation to integrate over each interval to compute the probability of possible number of children, the children probability distribution $\bar{C} = \bar{C}(F)$ is created. Thus, $P(\bar{C} = k)$ is the probability of $k$ children for a given union in the family network $F$, which is given by

$$\int_{k-0.5}^{k+0.5} \hat{g}(x; h)dx.$$



(a) $U$ (b) $\hat{U}$

Figure 5.2: Left: The discrete distribution of finite union distances $U$ for the Tikopia family network is shown. Right: Using the KDE to estimate the PDF of the union distance distribution, we interpolate any missing distances to create the probability distribution $\hat{U}$ by integrating the fitted curve shown.

## 5.3 Adding Generations to Network

After establishing the initial generation $g_0$ of the graph with the distance matrix $D_0(G)$ (as described in Section 5.2), unions are formed and children are added to create the next generation. However, before unions are formed among the current generation of vertices, more vertices are added to the graph using the probability of non-connected vertices $p_\infty$ (see Figure 5.1c). These added vertices form unions with randomly selected vertices from the current generation by creating a new union edge, each resulting in an infinite union distance. Each infinite-distance union is added to a list of new unions. Then, pairs of the remaining vertices in the current generation are selected based on the probability of their distance in $\bar{U}$ and a new union edge is created between each selected pair until the number of vertices in the list of new unions matches the fraction $p_{union}$. Because of this probability $p_{union}$, not every vertex will be paired as a union.

Once unions are formed, each union is randomly assigned a number of children based on the probability distribution $\bar{C}$. Vertices for these children are created and linked to each parent in the union via a parent-child edge. We note that not every union will have children, since $k = 0$ children is a possibility.

After all the parent-child relationships are established in the next generation, the distances between all vertices in the network are updated in the distance matrix $D_\mu(G)$, where

$0 < \mu < g$. For example, after the vertices in $g_0$ form unions and have children, the generation $g_1$ is established and the distance matrix $D_0(G)$ is updated and becomes $D_1(G)$. This updated matrix includes all the $n_0$ vertices from $g_0$ and the added non-connected vertices and the total number of added children. Thus, $D_1(G)$ is an $n_1 \times n_1$ matrix, where $n_1$ is the total number of nodes in the graph $G$ after the generation $g_1$.

Forming unions, adding children, and updating the distance matrix is repeated until all $g$ generations are created. Thus, the graph $G = G(F)$ becomes the modeled family network $M_F = (V_F, E_F)$ after $g$ generations.

# Chapter 6. Data

To test our model, we use the Tikopia and San Marino family networks from [4]. The Tikopia family network consists of 294 individuals from the island of Tikopia in Polynesia. As explained in Chapter 2, the San Marino network consists of 28,586 individuals from the Republic of San Marino, enclaved by Italy, from the 15th to the end of the 19th century. The finite union distance and children distributions for the Tikopia family network are shown in red in the bottom right corner of Figures 7.2 and 7.3 respectively. The Tikopia family network also has $d_\infty = 47$ infinite-distance unions, the probability of forming a union $p_{union} = 0.65$, and the probability of forming an infinite union $p_\infty = 0.21$. The finite union distance and children distributions for the San Marino family network are shown in red in the bottom right corner of Figures 7.8 and 7.9 respectively. The San Marino family network also has $d_\infty = 4,316$ infinite-distance unions, the probability of forming a union $p_{union} = 0.57$, and the probability of forming an infinite union $p_\infty = 0.17$.

One can think of the collection of genealogical data as going backwards in time. For example, to consider your own genealogical data, you start with yourself and trace back through your parents, grandparents, great-grandparents, etc. for as far back as possible. Once the data is collected, you can reverse this process and go forward in time from your ancestors to you. The model we propose goes forward in time. Starting with a specified number of individuals $n_0$ for the initial generation $g_0$, our model adds more individuals to the network with each generation for $g$ generations.

The combination of the parameters in Table 5.1 determine if the family network we create will continue indefinitely as $g \to \infty$ or die out. If the family eventually dies out, then the parameter $n_0$ effectively determines the size of the family network we create as $g \to \infty$. The larger $n_0$, the larger the resulting family is, on average. For the Tikopia family network, we found that this family eventually dies out after some number of generations $g_{max} = g_{max}(n_0)$ depending on the initial population $n_0$. After some experimentation, we found that $n_0 = 100$ is approximately the correct size at which to initialize our model to grow a network roughly the size of the actual Tikopia family network. We use $n_0 = 100$ to model the San Marino family network as well.

As $g_{max} = g_{max}(n_0)$, the number of $g$ generations the model will run for is dependent on the size of the original family network and the modeled family network. The model algorithm has a computational and spacial complexity of $O(n^2)$. Thus, as the modeled network gets bigger with each generation, the model requires more time to make the necessary computations. The parameter $g$ is set to $g = 14$ for the Tikopia and San Marino family networks. This number $g = 14$ ensures that both the model of the Tikopia and San Marino family networks grow as much as possible. For example, the Tikopia family network is relatively small and will eventually stop growing after some number of $g_{max}$ generations due to the unavailability of union candidates. After several realizations of the model, we found that the modeled Tikopia family network grows to a maximum of $g_{max} = 11$ generations if $n_0 = 100$. However, on average, the modeled network will grow for an average of $g_{max} = 9$ generations, which means that even when the model continues to run for $g = 14$ generations, the network will stop growing around 9 generations. Contrast this to the San Marino network that is much larger and appears to grow arbitrarily large as $g \to \infty$. The computational complexity prevents the modeled San Marino network from growing more than $g = 12$ generations, on average. That is, the San Marino network can grow to only an average of

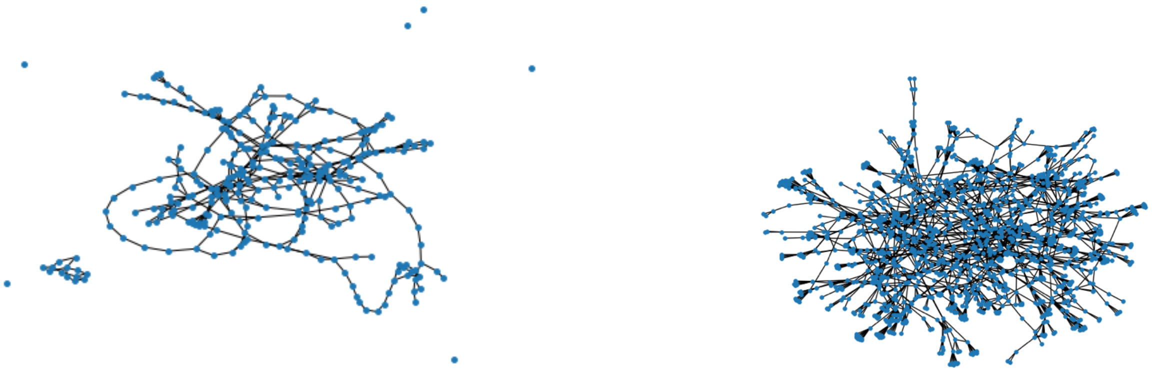
9,000 vertices before our simulations terminate. Thus, using a larger $g$ ensures that the network grows as much as possible.

It is important to note that the family network data we consider does not include a strict notion of generations nor any temporal factors besides parent-child relations. That is, there is no immediate way to know which unions and children are from certain generations. In some sense, the notion of a generation is a useful convention. Our model takes a simplified approach at generating family networks by not permitting intergenerational coupling or unions. We also limit each individual or vertex to forming at most one union, and only those unions can have children. Nonetheless, our results show that these simplifications do not deter our model from creating what appear to be realistic family networks (see Chapter 7).

In addition to the limitations of the computational and spacial complexity of the model, the algorithm used to compute persistence intervals is also limited. The program we used, called *Ripser* [21], has a computational and spacial complexity of $O((n+m)^3)$ where $n$ is the number of individuals and $m$ is the number of edges in a network. The number $n+m$ is the number of simplices in the network. In the Tikopia family network, $n = 294$, $m = 441$, and there are $n + m = 735$ simplices. In the San Marino family network, $n = 28,586$; $m = 51,446$; and there are $n + m = 80,032$ simplices.

Given our computational capabilities, the San Marino network is much too large for Ripser to compute persistence intervals. Thus, a *sampled* subgraph of the network is used to represent the original data. Starting from a random vertex of degree 1 in the San Marino network, we used a breadth-first search (BFS) to generate a *sampled network* that has $1,502 + 2,564 = 4,066$ simplices. Here, we chose a vertex of low degree to get at least some vertices at the boundary of the sampled network and some vertices in the interior. Figure 6.1 right shows this sampled network.

There are some drawbacks to using BFS-sampled networks instead of the complete network. Sampling a network in this way potentially creates a lot of *dead ends* in the resulting subgraph. That is, some cycles from the original network are severed in the sampling pro-



Tikopia Family Network

Sampled San Marino Family Network

Figure 6.1: Left: The Tikopia Family network is shown. This network has 294 vertices, 441 edges, and 735 simplices. Right: The sampled network of the San Marino family network is shown. This sampled network has 1,502 vertices; 2,564 edges; and 4,066 simplices.

cess. Additionally, this sampling technique does not take into account any communities found within the original network. Thus, the resulting structure of the sampled network is limited and potentially missing key factors of the original network. We further address how these issues affect our results in Section 7.2 and Section 7.3.

# Chapter 7. Results

In this chapter, we measure how similar real-world family networks and modeled family networks are by using target distributions, persistence curves, and bottleneck distances as described in Chapter 4. For each real-world family network, the Tikopia and San Marino networks, we made 10 realizations of the model to create several modeled family networks. We summarize our findings below.

## 7.1 Tikopia Family Network

A realization of a modeled Tikopia family network is shown in Figure 7.1, and demonstrates how the network grows with each generation, while emphasizing the largest connected component. As discussed, the driving force of our model is the union distance and children

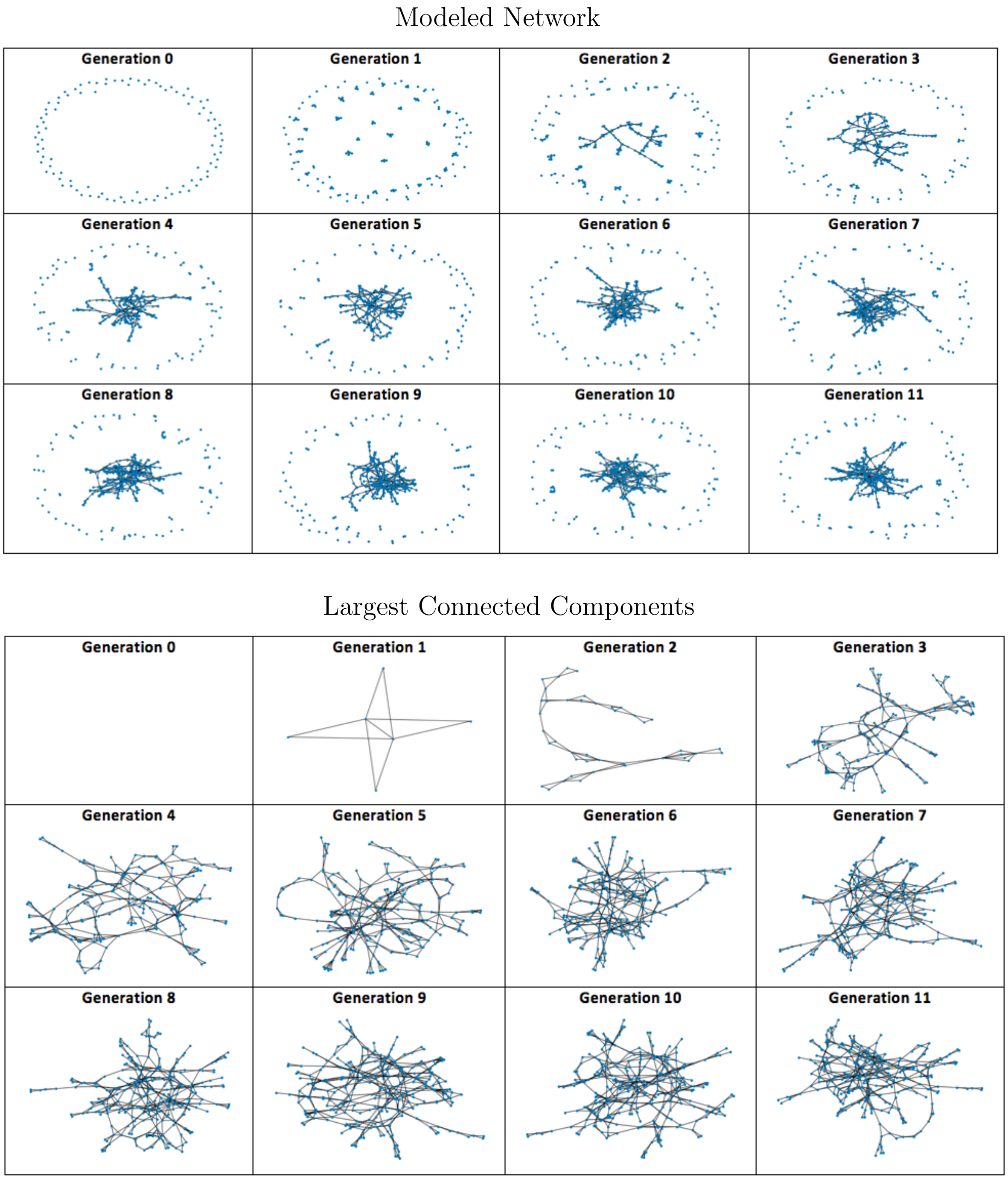


Figure 7.1: A modeled Tikopia family network at each generation is shown in the first grid. The second grid shows the largest connected component of the modeled network at each generation.

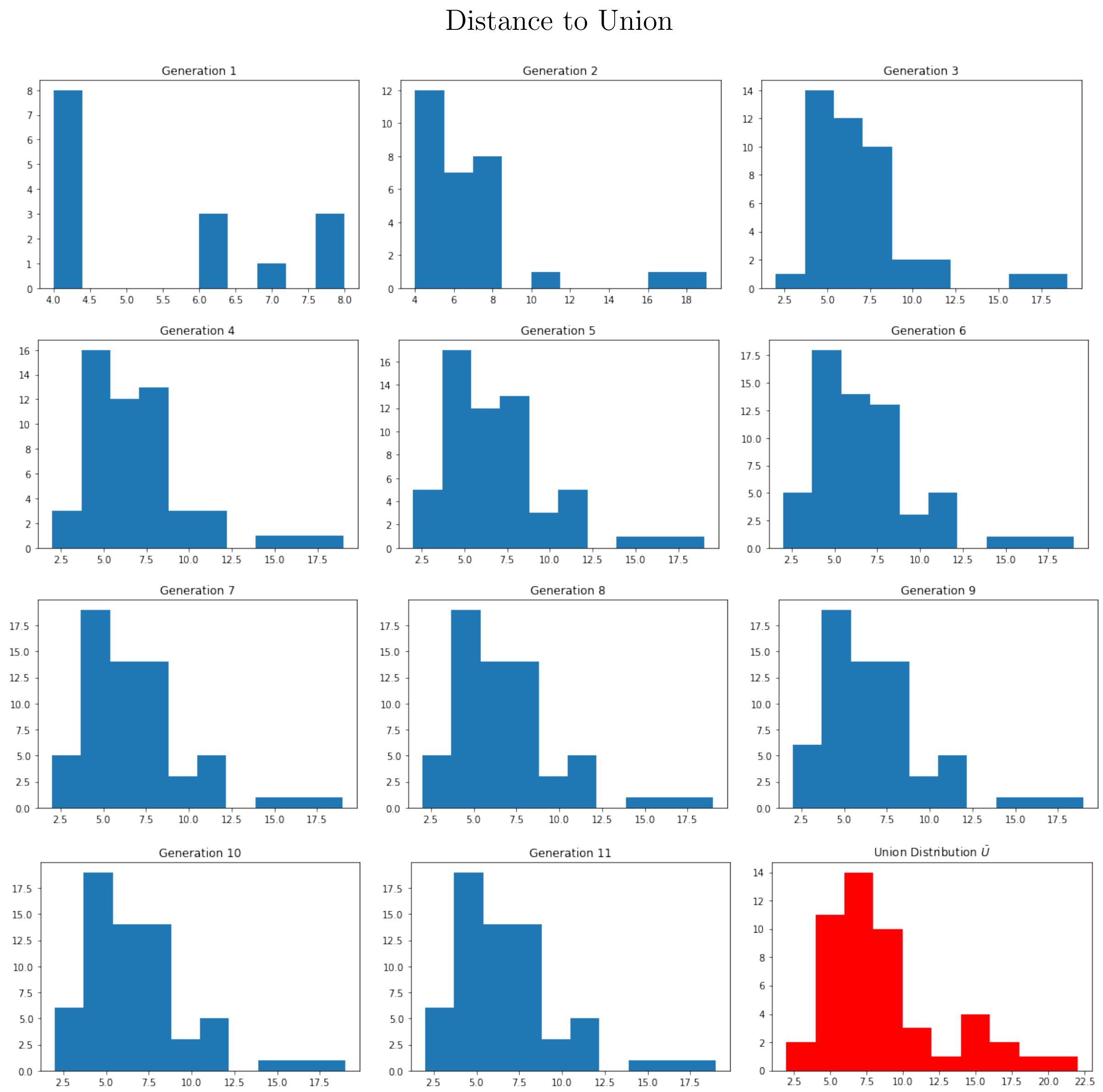


Figure 7.2: The union distribution for each generation of a modeled Tikopia Family network and the original union distribution $\bar{U}$ of the Tikopia Family network (bottom right corner) is shown.

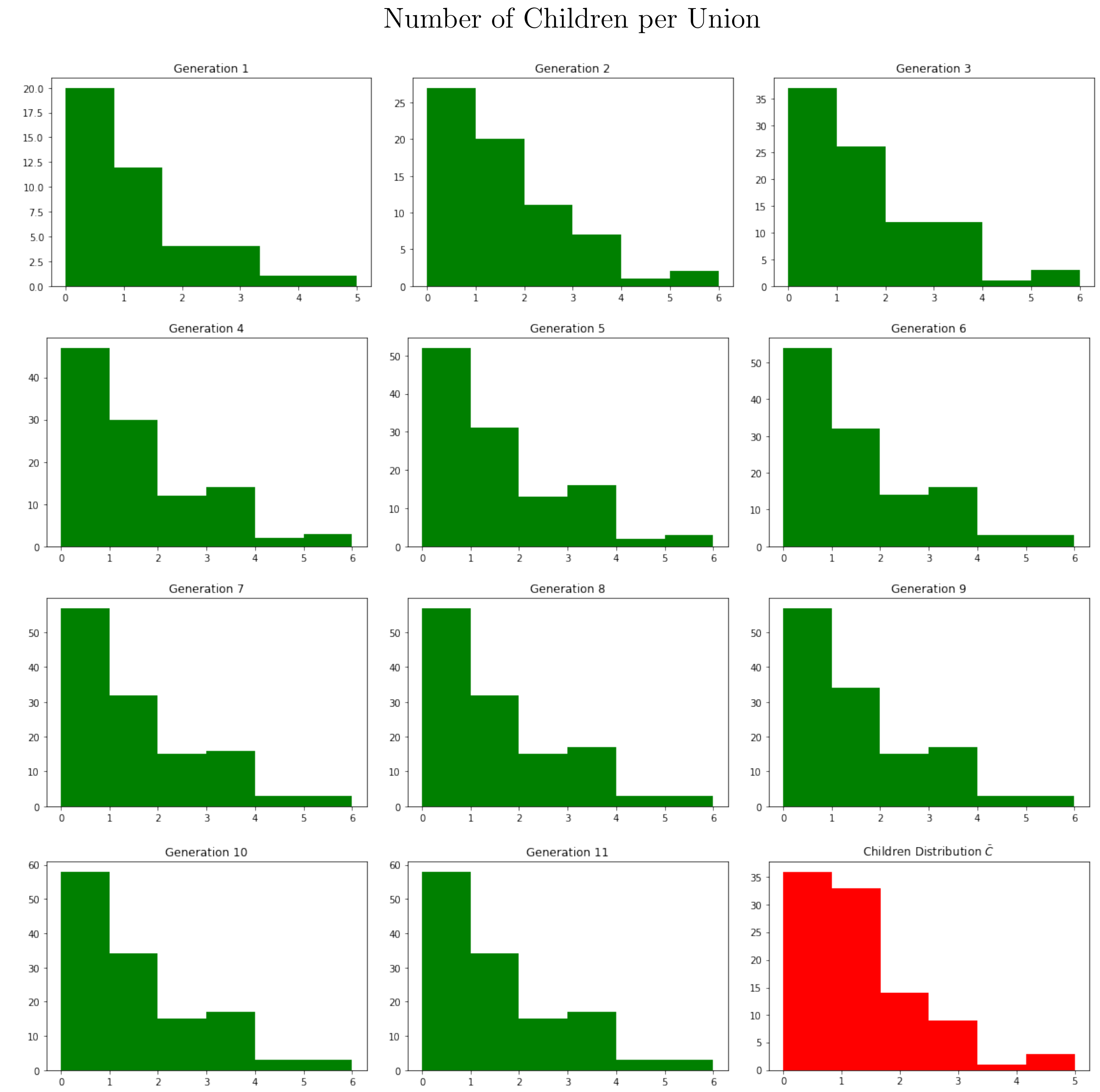


Figure 7.3: The children distribution for each generation of a modeled Tikopia Family network and the original children distribution $\bar{C}$ of the Tikopia Family network (bottom right corner) is shown.

distributions. These distributions determine how the modeled network grows. Figures 7.2 and 7.3 show the finite union distance distribution and children distribution of each generation of a modeled Tikopia family network. At each generation, these distributions roughly tend towards the target distributions (shown in red), indicating that the model correctly utilizes the various probability parameters. These three figures show that the modeled network is similar, but not identical, to the original Tikopia family network. Persistence homology equips us with mathematical tools to understand the extent of these similarities and differences.

Figure 7.4 shows how the persistence curves of a modeled Tikopia family network change with each generation. Persistence curves represent persistence intervals, which describe the different cycles of a network (see Section 4.2). Most cycles in family networks come from a combination of ancestor (parent-child) relationships and union relationships. Thus, the beginning stages of the modeled network, specifically, the initial generation $g_0$ through the second generation $g_2$, do not have enough height for ancestry relations. So, the persistence diagrams are computed entirely on union relationships, which means there are not enough persistence intervals to form nontrivial persistence curves. As such, the persistence curves in Figure 7.4 begin with generation $g_3$ and continue until the last generation of the modeled network, generation $g_{11}$. We see that with each generation, the shape of the persistence curve of the modeled network becomes more similar to the largest connected component of the original family network curve. We use the largest connected component to focus on larger network cycles. The other components of the original network are extraneous as they contain nontrivial small cycles that do not provide much information to the overall topology of the network (see left of Figure 6.1).

The biggest difference between the generational curves and the original family network curve is the number of length 2 intervals. Length 2 intervals represent cycles of length 4-8, approximately. We find the approximate length of a cycle by multiplying the persistence interval length by 3. These cycles of this particular size appear in a family network when

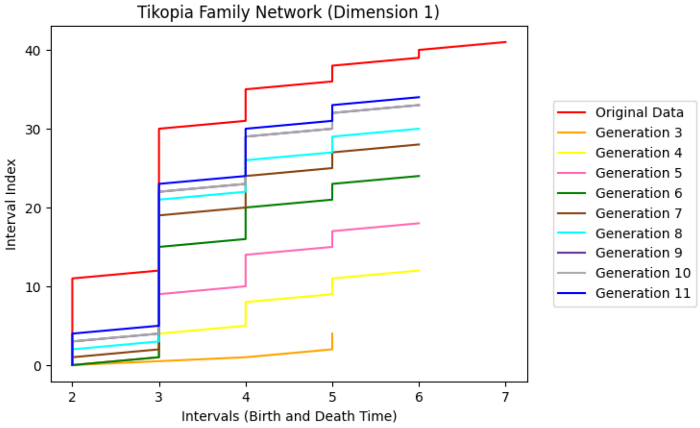


Figure 7.4: The persistence curves for dimension 1 of the original Tikopia family network (Original Data) and the curves of each generation of a modeled Tikopia family network are shown.

siblings form unions with other siblings (i.e., double cousins) and when a couple that does not share a union edge has children. Because our model restricts $\bar{C}$ children to couples who share a union edge and because the event of double-cousins has a small probability, our modeled networks will likely have fewer length 2 persistence intervals than the original network. This, in turn, signifies a good local fit of our model. Length 3 intervals are essentially the next generation of the aforementioned phenomenon since union shortcuts will still be missing at this point. Because of that, our modeled networks will also have fewer length 3 persistence intervals than the original network.

Larger persistence intervals are not affected by our model's union assumption because of *stability*. That is, persistent homology has a noise factor built in, so if there is only a single edge missing between two networks, then their persistence should be equal up to a left or right shift of 1. Correcting for these "small interval" factors, one can see that the shape of the step curves past length 4 are essentially the same for the modeled and the original network, indicating a very good fit between the global structure of our model and the original Tikopia family network.

The difference in number of length 2 intervals is further illustrated in Figure 7.5, which shows the persistence curves for 10 different realizations of a modeled Tikopia family network.

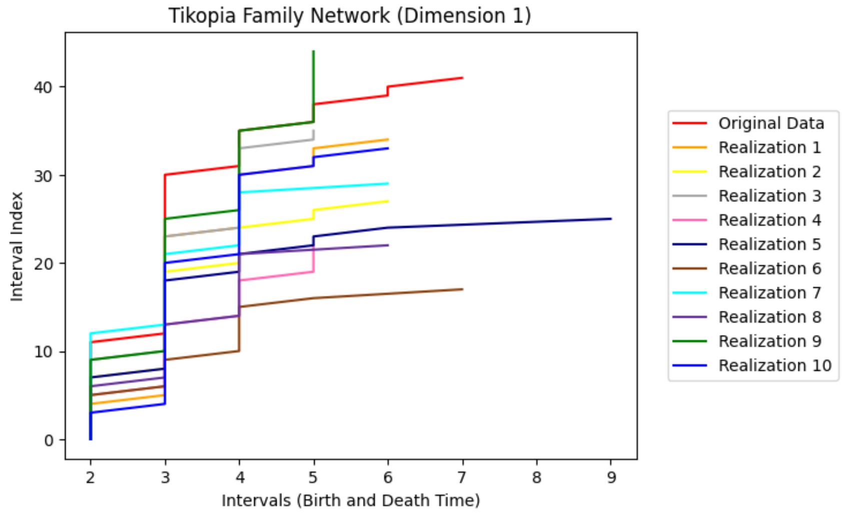


Figure 7.5: The persistence curves for dimension 1 of the original Tikopia family network (Original Data) and the curves of 10 realizations of a modeled Tikopia family network are shown.

Taking the last generation of each realization, we compute the persistence diagrams and corresponding persistence curves and plot them with the largest connected component of the original network's persistence curves. If we were to shift the curves of the realizations up to remedy the difference in length 2 intervals, they would more closely follow the original network's curves. This shifting would also show that some realizations' curves are a little above and a little below the original network's curve, demonstrating the randomization of our model around the “true” network.

After finding the persistence curves of 10 realizations of the model, we compute the bottleneck distance at each generation with the original Tikopia family network for each realization. Figure 7.6 (right) shows how the average bottleneck distance changes as the network grows with each generation. To have a bottleneck distance less than 1, the compared networks would essentially be the same. Thus, if there is any difference between the original and modeled networks, the bottleneck distance is at least 1. In Figure 7.6 (right) we see that the first three generations have an average bottleneck distance of 3. This is because the modeled network does not have ancestry-union complexity yet. After generation $g_3$, the

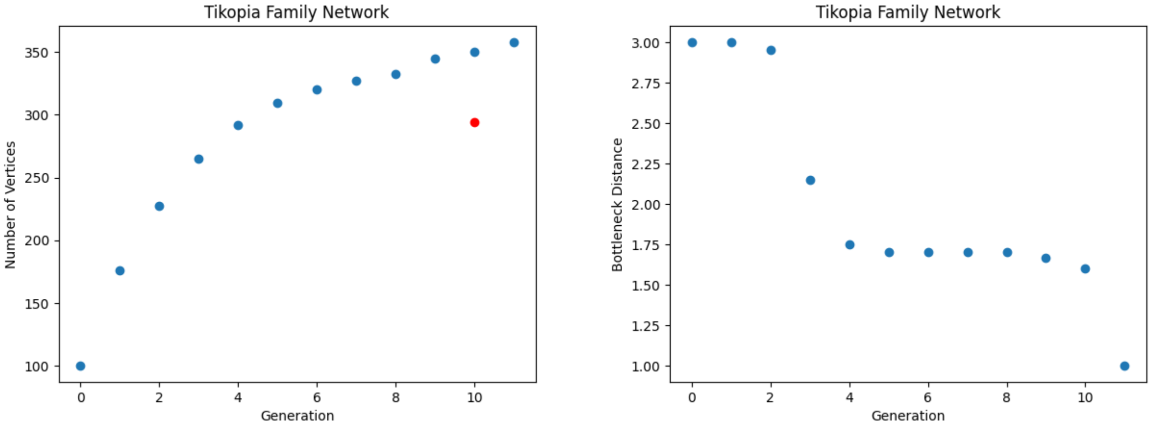


Figure 7.6: Left: The average size of 10 realizations of the modeled Tikopia family network at each generation is shown in blue. The size of the original Tikopia family network is shown in red (note, the data does not indicate how many generations are in the network, so plotting the point at generation $g_{10}$ was chosen arbitrarily). Right: The average bottleneck distance after 10 realizations of each generation between the original Tikopia family network and the modeled Tikopia Family network is shown.

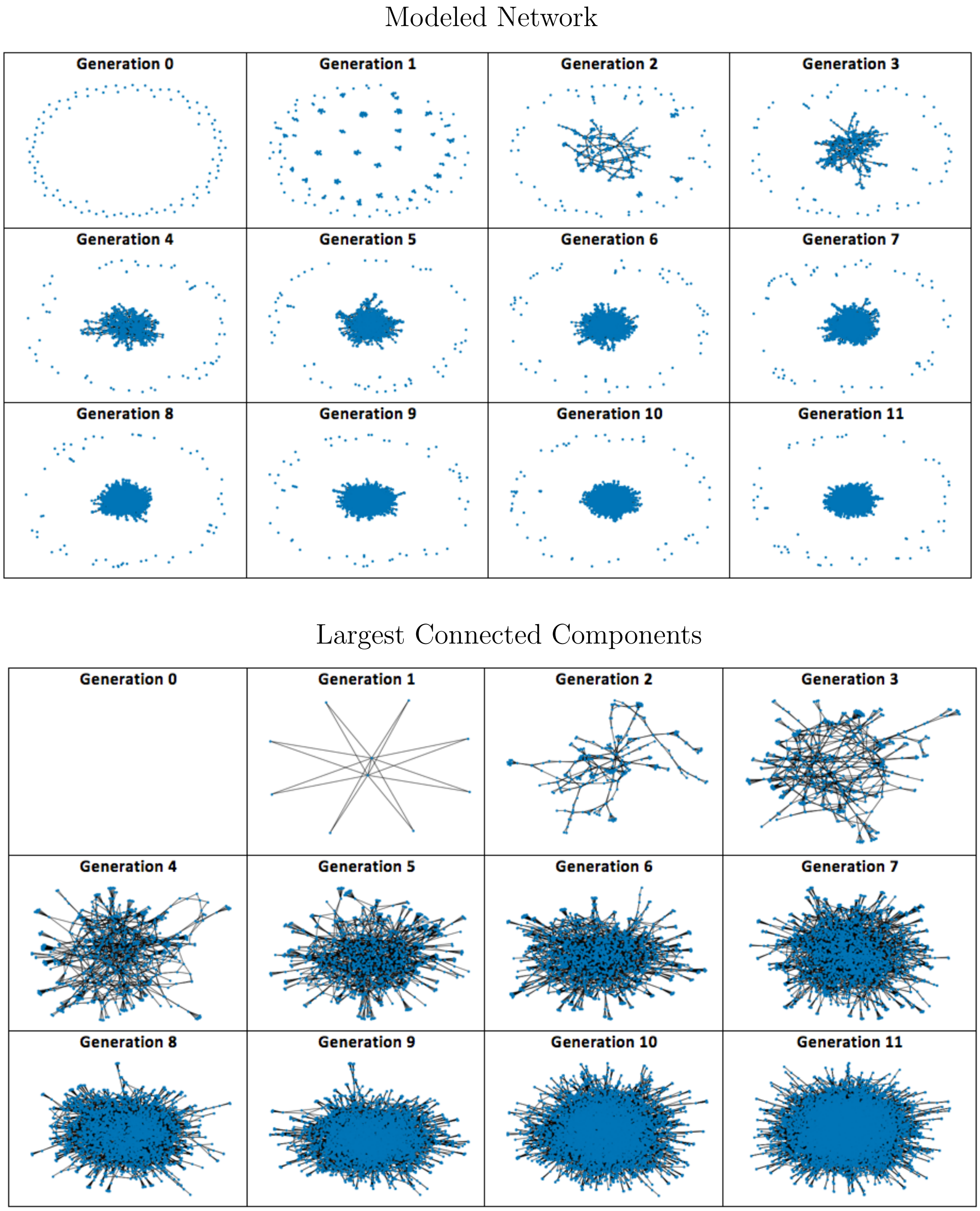
complexity of the network increases and the average bottleneck distance decreases. Realizations of the model that made it past 8 generations become more and more similar to the original Tikopia family network as seen by the continued decrease in bottleneck distance. Thus, if a modeled network reaches later generations, it will be more similar to the original Tikopia network. Modeled networks that do not have enough complexity, and in turn cannot continue to grow for more generations, will not be as realistic as those networks that do. Figure 7.6 (left) shows the average number of vertices in the network at each generation, including the number of vertices in the original family network (shown in red). It is interesting to note that even though the average size of the modeled family network exceeds the size of the original family network, especially in later generations, the bottleneck distances tell us that their underlying structure is similar.

## 7.2 San Marino Family Network

The sheer size of the San Marino family network is quite a contrast to the Tikopia family network. A realization of a modeled San Marino family network is shown in Figure 7.7. Comparing this to the realization of the Tikopia family network in Figure 7.1, we see that the modeled San Marino family network grows a lot more in the eleven generations than the modeled Tikopia family network. As explained in Chapter 6, computational limits prevent the modeled San Marino family network from growing to the same size as the original family network. On average, the modeled San Marino family network grows to about 9,000 vertices, whereas the original San Marino family network has about 28,500 vertices. Despite this, Figures 7.8 and 7.9 show that the finite union distance distribution and children distribution are roughly tending towards the target distributions (shown in red). Assuming that the modeled San Marino network grew large enough, we would expect it to be similar to the original San Marino family network.

As discussed in Chapter 6, computing persistence curves is too computationally expensive for large family networks. Thus, after 10 different realizations of a modeled San Marino family network, we sample each resulting network to get 10 sampled, modeled San Marino family networks of size 1,200 vertices, on average. We compute the persistence diagrams and bottleneck distance of each realization with the original, sampled San Marino family network as shown in Figure 7.10 (right). Section 4.1 and 7.1 explain that if networks are similar, their bottleneck distance must be small; however, the converse is not true. Thus, we also compute the persistence curves of the 10 sampled, modeled San Marino family networks and plot them with the persistence curve of the original, sampled San Marino network to better understand how similar the sampled, modeled networks are to the original, sampled network (see left plot of Figure 7.10).

In Figure 7.10 (left), the persistence curves of each realization resemble persistence curves of family networks. Nonetheless, the cycles found in the sampled, modeled San Marino family networks differ from the original, sampled San Marino family network. When sampling



Modeled Network

Largest Connected Components

Figure 7.7: A modeled San Marino family network at each generation is shown in the first grid. The second grid shows the largest connected component of the modeled network at each generation.

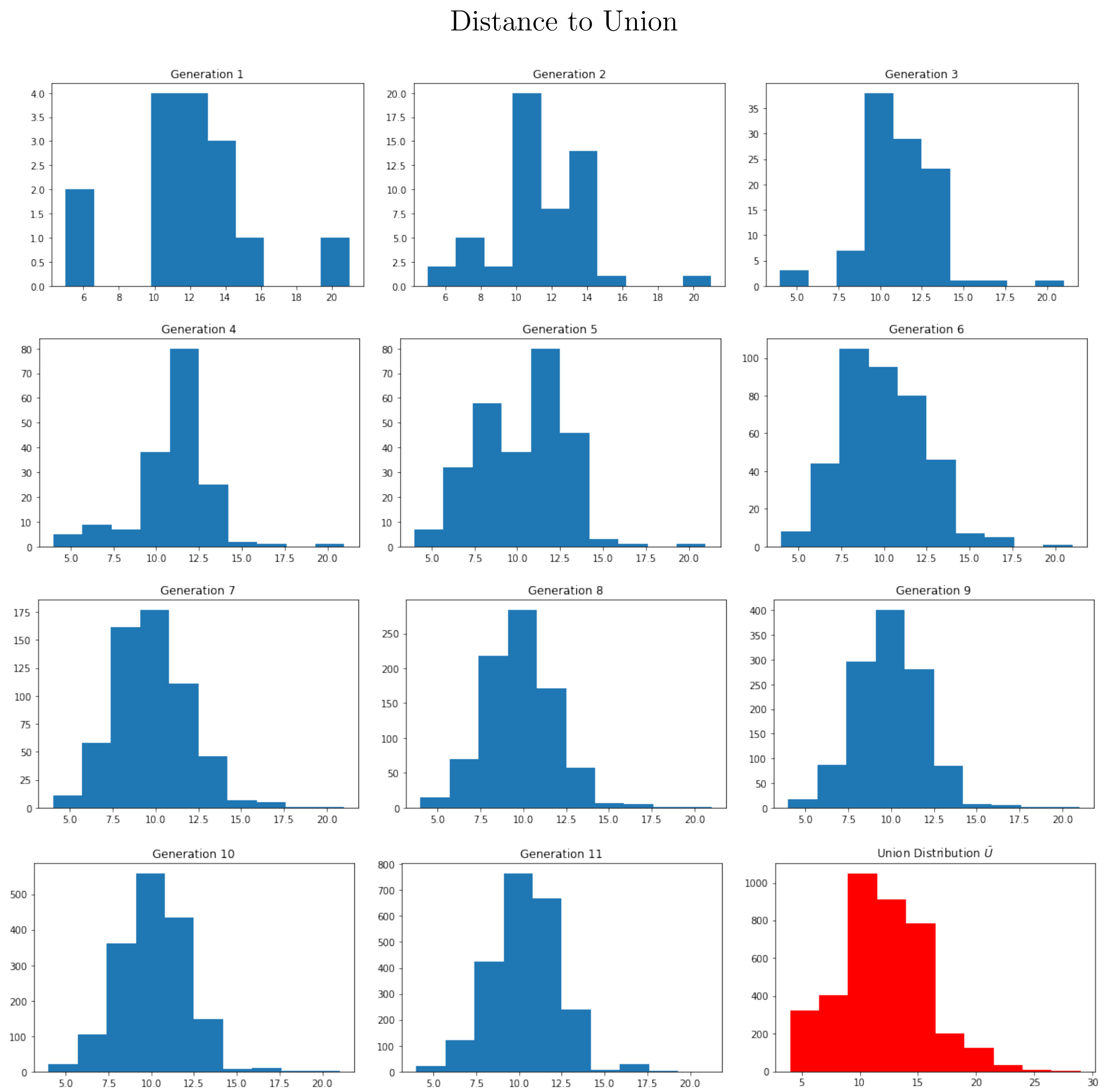


Figure 7.8: The union distribution for each generation of a modeled San Marino Family network and the original union distribution $\bar{U}$ of the San Marino Family network (bottom right corner) is shown.

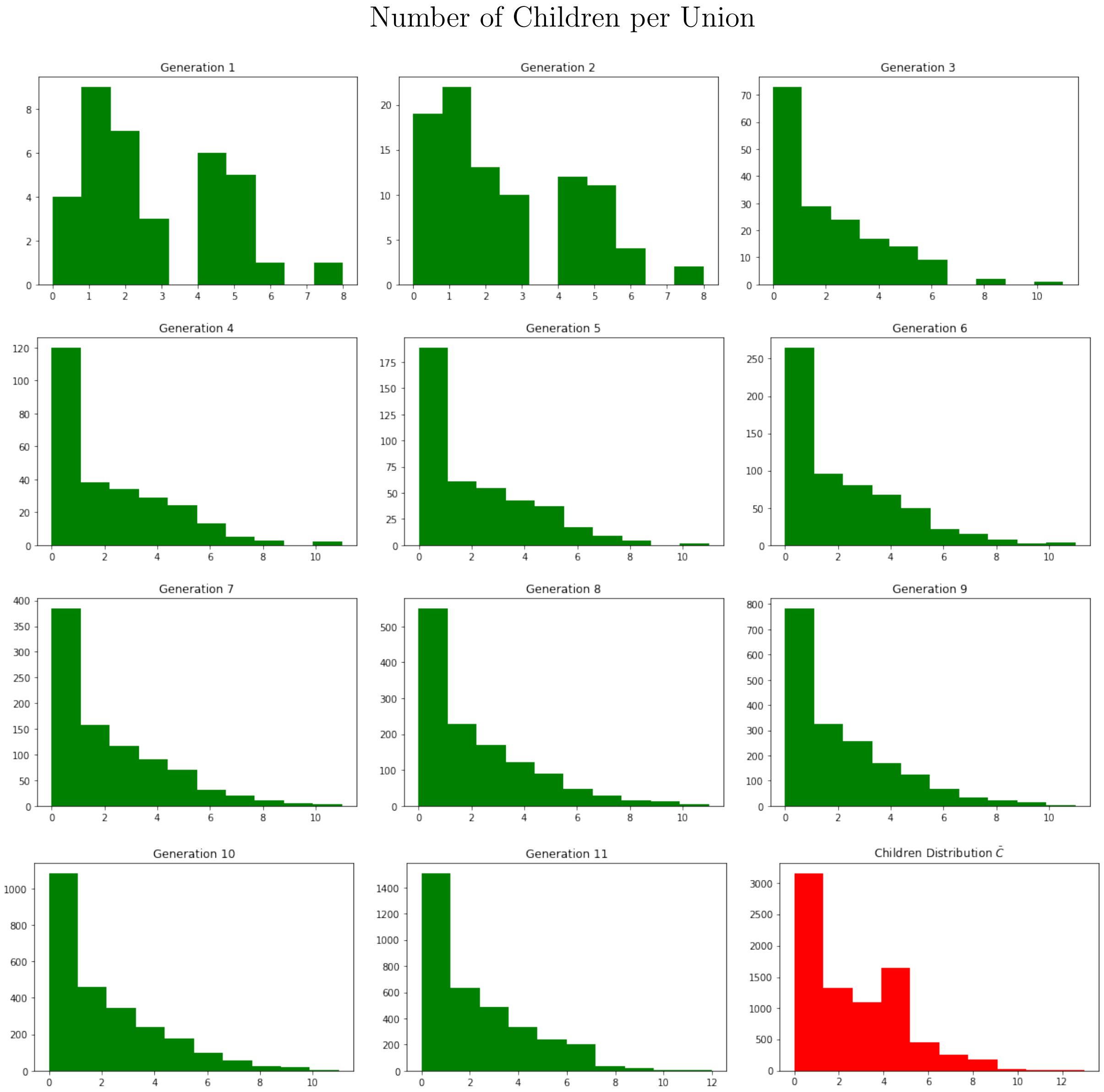


Figure 7.9: The children distribution for each generation of a modeled San Marino Family network and the original children distribution $\bar{C}$ of the San Marino Family network (bottom right corner) is shown.

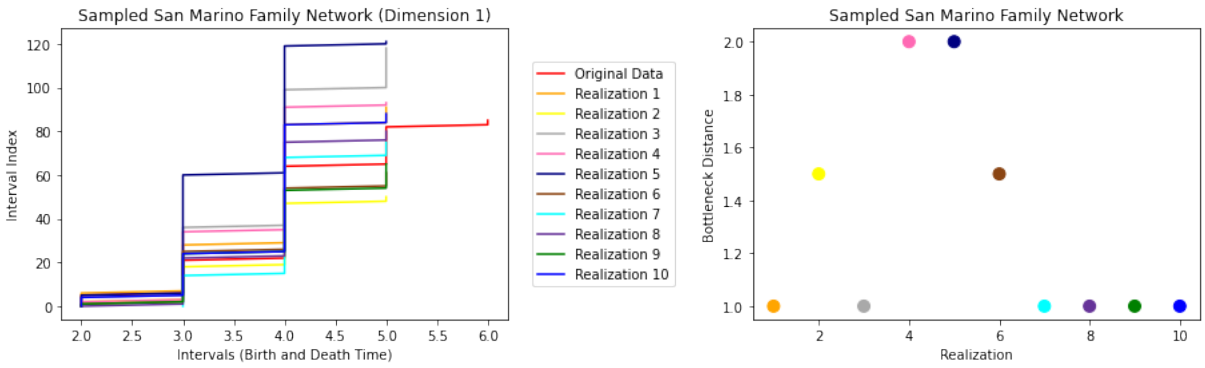


Figure 7.10: Left: The persistence curves for dimension 1 of the original, sampled San Marino family network (Original Data) and the curves of 10 realizations of a sampled, modeled San Marino family network at the last generation are shown. Right: The bottleneck distances between the original, sampled San Marino family network and the 10 realizations of the sampled, modeled San Marino family network at the last generation are shown.

networks, cycles from the original network are cut off and communities within the network are not accounted for. As such, we see that sampling has a negative impact on measuring the accuracy of our model. Thus, we cannot adequately conclude how similar the modeled San Marino family networks are to the original San Marino network.

## 7.3 Conclusion and Future Work

Our results suggest that (i) the distance at which unions form and (ii) the number of children each union has are fundamental features that determine the structure of family networks. Our model utilizes these features to create realistic family networks. However, we recognize that there are limitations to these modeled family networks.

As explained in Chapter 6, our model simplifies real-world events by not permitting intergenerational coupling or unions. More examination of generational factors would lead to a better understanding of how intergenerational coupling occurs, which would help improve our model. Additionally, it may be beneficial to allow multiple unions or separation of unions throughout time. That is, allow an individual to form a union, separate from that union, and then form another union. Or, allow unions to separate without the formation of another union. This would potentially require intergenerational coupling since separation and reformation would possibly not occur in the same generation.

Moreover, the assignment of children could also be improved. The current model only allows union pairs to have children; thus limiting the addition of more people to the family network. Our model could also consider including children of the same immediate family to be from more than one generation. In the real world, it is possible for a union to have a large age gap between children. Our current model does not necessarily take this into account due to the data's limitation of temporal factors.

Ideally, we would want to model large family networks. Generally, larger networks better capture real-world events and are more complete. The computational limitations of our model and Ripser make studying large family networks difficult. Sampling large networks to mitigate these limitations has several drawbacks as explained in Chapter 6 and as seen in Section 7.2. Further study could reveal that BFS is not the best possible algorithm for sampling networks, and that other algorithms would reduce some of the drawbacks mentioned. Alternately, we could explore other computational capabilities to handle large family networks. For example, *Ripserer* [22] is a pure Julia implementation of the Ripser algorithm, allowing for much faster computations than the Python package of the algorithm we used. This alternate program would enable a better study of the San Marino family network and other larger networks without needing to sample the network.

Our model establishes a foundation for generating modeled family networks. Further study would lead to even better results and even more insights to the characteristics of human family networks.

# Bibliography

[1] Vanessa Robins, Mohammad Saadatfar, Olaf Delgado-Friedrichs, Adrian P. Sheppard. Percolating Length Scales from Topological Persistence Analysis of Micro-CT Images of Porous Materials, *Water Resources Research*, Volume 52, Number 1, pp. 315–329 (2016).

[2] H. Lee, H. Kang, M. K. Chung, B.-N. Kim, D. S. Lee. Persistent Brain Network Homology from the Perspective of Dendrogram, *IEEE Transactions of Medical Imaging*, Volume 31, Number 12, pp. 2267–2277 (2012).

[3] Mattia G. Bergomi, Adriano Barate, Barbara Di Fabio. Towards a Topological Fingerprint of Music, *Proceedings of the 6th International Workshop on Computational Topology in Image Context*, Volume 9667, pp. 88–100 (2016).

[4] Family Network data https://www.kinsources.net/browser/datasets.xhtml (last accessed Jan. 2021).

[5] Residence hall social network data http://konect.cc/networks/moreno_oz/ (last accessed Jul. 1, 2021).

[6] Deezer Social Network data https://snap.stanford.edu/data/feather-deezer-social.html (last accessed May 10, 2021).

[7] E. Malmi, A. Gionis, and A. Solin. *Computationally inferred genealogical networks uncover long-term trends in assortative mating*, in Proceedings of the 2018 World Wide Web Conference WWW 2018, Lyon, France, April 23-27, 2018, pp. 883–892. Available at http://doi.acm.org/10.1145/3178876.3186136.

[8] G. Bloothooft, P. Christen, K. Mandemakers, M. Schraagen. *Population Reconstruction*, Springer, 2015.

[9] J. Greenwood, N. Guner, G. Kocharkov, C. Santos. Marry your like: Assortative mating and income inequality, *American Economic Review*, Volume 104, Issue 5, pp. 348–353 (2014).

[10] K. Hamberger, M. Houseman, D. R. White. *Kinship, class, and community*, in The SAGE Handbook of Social Network Analysis, J. P. Scott and P. J. Carrington, eds., Sage Publications Ltd., pp. 129–147 (2011).

[11] P. Hage and F. Harary. *Structural models in anthropology*, Cambridge University Press, Cambridge, 1983.

[12] Joseph T. Chang. Recent common ancestors of all present-day individuals, *Advances in Applied Probability*, Volume 31, Issue 4, pp. 1002–1026 (1999).

[13] D. L. T. Rohde, S. Olson, J. T. Chang. Modelling the recent common ancestry of all living human, *Nature (London)*, Volume 431, Issue 7008, pp. 562–566 (2004).

[14] H. J. and C. A. Machado. The study of structured populations–new hope for a difficult and divided science, *Nature Reviews. Genetics*, Volume 4, Issue 7, pp. 535–543 (2003).

[15] H. Kannan, E. Saucan, I. Roy, A. Samal. Persistent homology of unweighted complex networks via discrete Morse theory, *Scientific Reports*, Volume 9, pp. 1–18 (2019).

[16] C. J. Carstens and K. J. Horadam. Persistent Homology of Collaboration Networks, *Mathematical Problems in Engineering*, Volume 2013, Article ID 815035, 7 pages.

[17] Afra Zomorodian, Gunnar Carlsson. Computing persistent homology, *Discrete & Computational Geometry*, Volume 33, Issue 2, pp. 249–274 (2005).

[18] D. Cohen-Steiner, H. Edelsbrunner, J. Harer. Stability of persistence diagrams, *Discrete & Computational Geometry*, Volume 37, pp. 103–120 (2007).

[19] M. P. Wand and M. C. Jones, *Kernel Smoothing*, Chapman & Hall, 1995.

[20] Kernel Density Estimation https://scikit-learn.org/stable/modules/density.html (last accessed Oct. 27, 2021).

[21] Ripser Python package https://anaconda.org/conda-forge/ripser (last accessed Oct. 4, 2021).

[22] Ripserer Julia package https://mtsch.github.io/Ripserer.jl/dev/ (last accessed Nov. 17, 2021).