Brigham Young University
BYU ScholarsArchive

Theses and Dissertations

2023-06-12

The Target Model for Genealogical Networks

Kolton Baldwin
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

 Part of the Physical Sciences and Mathematics Commons

BYU ScholarsArchive Citation

Baldwin, Kolton, "The Target Model for Genealogical Networks" (2023). *Theses and Dissertations*. 9985.
https://scholarsarchive.byu.edu/etd/9985

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.

The Target Model for Genealogical Networks

Kolton Baldwin

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
Zachary Boyd
Mark Kempton

Department of Mathematics

Brigham Young University

Copyright © 2023 Kolton Baldwin

All Rights Reserved

ABSTRACT

The Target Model for Genealogical Networks

Kolton Baldwin
Department of Mathematics, BYU
Master of Science

Several large-scale projects including FamilySearch, Ancestry, BALSAC (University of Quebec), and others have gathered incredible amounts of genealogical data ranging from millions to billions of individuals. To study the structure of this data, we propose a model that generates a genealogical network based on real-world genealogical data using two key features: (i) geodesic distance between couples prior to union and (ii) the number of children per couple. The distribution of the distance to a couples' nearest common ancestor in an observed community captures the global scale at which biological cycles form in the underlying genealogical network. Similarly, the number of children per couple captures the local structure given by the degree distribution in the genealogical network. Constructing imitation data which approximates a real-world network's structure and growth rate is desirable for use in generalizable machine learning models. This model, which we refer to as the Target Model, provides a foundation for further work in predicting family network growth and structure.

Keywords: genealogical networks, distance to union, target model

Acknowledgements

I would like to thank the members of my committee: Dr. Benjamin Webb, Dr. Zachary Boyd, and Dr. Mark Kempton for their support and encouragement throughout my time as a graduate student. I would also like to thank the other members of our research group: Abby Jenkins, Rebecca Flores, Jordan Sheppard, Michael Okuda, Sukyoung Qwak, and Teayoun Kim for their support.

Contents

List of Tables

List of Figures

Chapter 1. Introduction

Genealogical research has grown from a dusty, closeted pass time to a big, multifaceted business. Genealogical data that was once transcribed and painstakingly recorded by hand has transformed into a booming digital industry. Domestically, the genealogy industry is valued at more than $8.5 billion (USD) and is expected to double in value in less than seven years [FutureWise report HC-1137]. While genealogical research in its own rights has been lauded as the "second most popular hobby in the U.S. after gardening," the industry includes far more than tracing pedigrees [1]. Some of the growth in genealogical interest is the increasing availability of genealogical data and some may be attributed to developments in technology. Consider the sheer size of FamilySearch's family tree. As of early 2023, its network spans more than 1.5 billion individuals. The FamilySearch digital record collection contains information for an additional twelve billion individuals whose marriage and family relationships have not yet been entered into their massive genealogical network [2]. Such diverse industries as medicine and economics have seen new horizons unfold when examining their respective disciplines through a family history lens [3, 4]. Many of these developments include the various applications of consumer genomics [4].

While several genealogical software companies have implemented data structures for users to record their research, such record-keeping structures are tools only. Other technological innovations reach beyond record keeping to investigate the actual structuring of genealogical networks. Some of these attempts have been made at automatically reconciling and joining disparate genealogical datasets [5] and some efforts have been made to automatically form genealogical networks from digitized documentation [6]. Of note, are studies concerning a search for a population's most recent common ancestor [7].

This thesis proposes that a genealogical network's structure is dominated by two features. First, how closely partners are related before union (e.g. marriage) and second, how many children each household has. To justify this, we construct a model that creates artificial

networks which mimic the structure of a real-world genealogical network. We then show that our modeled networks approximate both features of the real network.

The structure of this thesis is as follows. In Chapter 2, we describe the relationship between family networks and genealogical networks. In Chapter 3, we describe the main characteristics of genealogical networks and introduce the (i) distance to union and (ii) children per household distributions as characteristic and descriptive of a genealogical dataset. In Chapter 4, we introduce the Target Model for generating networks which approximate a given genealogical network using (i) and (ii). In Chapter 5, we propose a simplified variant of the Target Model which reduces the size of the networks produced but often at the expense of losing accuracy in recapturing global network structure. In Chapter 6, we discuss the advantages and disadvantages of each variation of the Target Model and we explain how differences in measuring technique affect the accuracy of the model.

Chapter 2. Family Networks and Genealogical Networks

Genealogical and family networks are similar but have important differences. Foremost, a *genealogical network* is a subset of a family network. While both types of networks trace familial relationships and have at least a relative temporal orientation, i.e. families grow as parents have children, but not as children have parents, genealogical and family networks differ in their scope and level of precision. A family network represents a complete set of connections—including extended family members, all births, and all unions—regardless of whether or not such connections were accurately or ever recorded. Additionally, a genealogical network is a curated collection of real-world family data which contains some of—but not necessarily all of—the information that the underlying ground-truth family network contains. For example, American genealogists may find that children who were born and died between sequential U.S. Censuses (held every ten years in the U.S.) are more easily missed in their research. Such a child would be represented in the ground-truth family network—he

or she really was born to their parents—but such a child may conceivably not be represented in an imperfectly-curated genealogical network.

We focus on genealogical networks, not only as a matter of feasibility, but as a necessary stepping stone to understanding the connections between genealogical networks and their underlying family networks. We show that the local and global structures of genealogical networks are largely determined by (i) the distribution of how closely partners are related and (ii) the distribution of the number of children per household. We hope that this model will eventually provide a way to understand and to measure the current and future completeness of our collective genealogical data.

A genealogical network has an underlying graph structure and can be represented by a graph $G = (V, E)$ which is comprised of a set of vertices $V = \{1, 2, ..., n\}$ which represent individuals and a set of edges E which represent familial relationships between these individuals. The relationship between vertices i and j is represented by an arc connecting i and j with edge $e_{ij} = e_{ji} \in E$ from i to j. Notably these edges are of two types. An edge either represents a union between two vertices or else it represents a parent-child relationship. The set of all unions is represented by E_u and the set of all parent-child relationships is represented by E_{pc}. No edge represents both a union and a parent-child relationship simultaneously, so that the set of edges is the disjoint union $E = E_{pc} \cup E_u$. Both $|V| < \infty$ and $|E| < \infty$, but vertices are not restricted in the number of edges they are connected to and a vertex may be connected to both union- and parent-child-type edges. Union edges are undirected and are unweighted. Parent-child edges are likewise unweighted but are directed from parent to child. To find a genealogical network's (i) distance to union distribution we search for two vertices' nearest common ancestor making careful use of this directed relation. Given a pair of vertices, their *nearest common ancestor* is the most-recent direct-line ancestor from which a path of parent-child edges may be followed to each of the two vertices. This nearest common ancestor has a biological path of parent-child edges to each of these two vertices and the combined length of these paths is not more than the combined path lengths from any

other common ancestor to each vertex in the pair. Parent-child edges allow travel forward in time but not backward as parent-child edges begin at a parent vertex and end at a child vertex.

The term union edge will describe any relationship which could result in children—including adopted children. For instance, union edges include marriages, but are not restricted thereto. If a person has married more than once, then they will have more than one union edge connected to them. Each union edge and its connected vertices will be treated separately as their own household. A *household* is comprised of exactly one union edge, both partners joined by that edge, and all children connected by a parent-child edge to either parent.

Chapter 3. Model Parameters and Outline

We propose the Target Model as a model of genealogical networks which approximates the structure and characteristics of a specific real-world genealogical network. Given a *target* real-world genealogical network $G = (V, E)$, we measure both the size and the interconnectedness of its individuals. As mentioned in the previous two chapters, we form probability distributions of two specific quantities: (i) the distance to nearest common ancestor per union $P_U(x)$ and (ii) the number of children per household $P_C(x)$. The distribution of children essentially dictates the network's degree distribution and so accounts for the local structure of the network. However, the distance to union distribution gives only a coarse-grained view of the global structure of the network. That is, two very different networks can share the same distance to union distribution. We use these probability distributions to create a network which approximates the original or *target* real-world network in its size, distribution of the number of children, and distance to union distribution. Whether or not these two distributions together constitute sufficient information to meaningfully recreate a genealogical network's structure is one of the main questions motivating this thesis.

Because our proposed model is stochastic, realizations of our model differ one from an-

other and from the original real-world network. Our model begins with an initial $n_0 > 0$ number of individuals in the network's 0th generation g_0. We then randomly draw from the distance to union distribution to determine a relative distance between each pair of these original individuals. Next we build out ancestral lines necessary to support these randomly-determined distances before proceeding to form unions based on these relative distances from among the possible pairings in g_0 (see Section 4.2). We then form the most likely pairings relative to $P_U(x)$, joining some fraction of individuals in g_0 with union edges in what we call a *finite-distance* unions. Then some still uncoupled individuals in g_0 are randomly selected to form a union with an individual who is not connected to our graph, in what we refer to as an *infinite-distance* union.

Once both finite- and infinite-distance unions are formed in g_0, then the (ii) number of children per union is created using the distribution from the original target network. These children constitute 1st generation g_1 and the model repeats by forming finite-distance and infinite-distance unions from among the vertices of g_1 before adding a new generation of children to each of the households in g_1. This creating of the following generations $g_2, g_3, ..., g_L$ continues until the total number of individuals in $g_0, g_1, ..., g_L$ exceeds some fixed number with the creation of the final or *last generation* g_L.

The purpose of the Target Model is to augment genealogical datasets with artificially-created networks which preserve characteristics of a specific genealogical network for use in predictive modeling. We later show that the Target Model produces networks whose distance to union and child distributions well approximate those of the original genealogical network, in addition to other similarities between our modeled networks and their real-world counterparts (see Chapter 6). This indicates that our two main feature distributions effectively characterize the structure of a genealogical network.

As mentioned, given a real-world genealogical network $G = (V, E)$, we measure several characteristics of its structure for use as parameters in constructing target models of the original genealogical network. This includes measurements such as number of vertices $|V| =$

n_{target} and number of union edges $|E_U| = m$ which will play a part in our model. However, network size alone insufficiently captures the complexity of the network's structure. Metrics such as the genealogical network's distributions of (i) distance before union $P_U(x)$ and (ii) children per household $P_C(x)$ are much more informative. The process of measuring a real-world genealogical network for each of these is quite straight-forward. This process is described in the following section.

3.1 Data

Our codebase for creating a model of a target network is designed for use with Pajek ore-graph data files (.paj extensions). However, any genealogical network with the required network structure could work [8]. For our model to operate a target genealogical network with the following characteristics must be supplied. First, the network must have an accompanying list of undirected union edges. Next, the network must have a list of directed parent-child edges, directed from parent to child. Finally, the network must have an average rate of union not more than one union per individual.

A pajek file encodes basic network information in three different portions. First, Pajek ore-graph files contain a list of vertices which represents each individual on a new line. (Pajek files may contain individuals' names and sexes here, but our model does not require this information.) Second, the list of vertices is followed by a list of parent-child edges. Each line in this section represents a new parent-child edge encoded in three numbers representing, respectively, the parent node, the child node, and the edge weight (in a Pajek ore-graph format all edges have a weight of one). Pajek formatting refers to parent-child type edges as arcs. Third, Pajek ore-graph files contain a list of union edges, which mimics the formatting of the list of parent-child edges. Three integers indicate the first spouse, the second spouse, and an edge weight respectively. Again, in a Pajek ore-graph format all edges have a weight of one. Pajek formatting refers to union edges as edges [9]. An example of a Pajek ore-graph file is shown in Table 3.1.

Contents and Format of Pajek Ore-graph data file (.paj extension)		
	Fomatting	Example Portion of Pajek (.paj) File
Vertices: For each vertex in the network, list index number, name, and sex.	**section header:** {*vertices} {no. vertices} **line contents:** {numerical vertex index no.} {'string vertex name'} {vertex sex} {new line}	*vertices 2588 1 'John' triangle 2 'Jane' ellipse 3 'James' triangle ⋮
Arcs: For each parent-child edge in the network, list the index number of the parent, the index number of the child, and the weight of the edge.	**section header:** {*arcs} **line contents:** {parent vertex} {child vertex} {edge weight}	*arcs 1 14 1 1 4 1 1 3 1 ⋮
Edges: For each union edge in the network, list the index number of the first partner vertex, the index number of the second partner, and the weight of the edge.	**section header:** {*edges} **line contents:** {partner vertex} {partner vertex} {edge weight}	*edges 1 2 1 3 2303 1 4 1886 1 ⋮

Table 3.1: Pajek (.paj) file formatting consists of a text file divided into three sections: vertices, arcs (parent-child egdges), and edges (union edges). Each vertex's sex is listed, with a corresponding shape: triangle for male, ellipse for female, and square for unknown.

Throughout this thesis we reference a collection of 105 genealogical datasets which are freely available online. These genealogical networks vary in their size and other characteristics (see Appendix A) [10].

3.2 Distance to Union Distribution

We collect the distribution of specific path lengths between the individuals connected by each union edge. In a graph $G = (V, E)$, a *path* of *length* $\ell \geq 1$ is a sequence of vertices in $(v_1, v_2, ..., v_\ell)$ with $v_i \in V$ for $i = 1, 2, ..., \ell$ such that each consecutive pair vertices is

connected by an edge $e_{j,j+1} \in E$ for $j = 1, 2, ..., \ell - 1$ and in which no vertex is listed multiple times [11]. The length of the shortest path between two vertices is defined as the *distance* between them.

In our search for the nearest common ancestor shared by a couple, we conduct a time-dependent *breadth-first search* (BFS) on a subgraph of G comprised of the full set of vertices but only over the parent-child edges of the graph, with the orientation of each edge reversed. That is, we do a BFS over the *biological subnetwork* of a genealogical network graph, with the orientation of the parent-child edges reversed. That is, on the network $G_{bio}^{-1} = (V, E_{pc}^{-1})$. Recall that the set of parent-child edges E_{pc} are directed from parent to child, so that the set of edges E_{pc}^{-1} denotes the same set, but with the head and tail of each edge reversed, i.e. edges in E_{pc}^{-1} are directed from child to parent. Our search algorithm makes careful use of the direction of each edge to ensure that we find a common direct-line ancestor, not a common descendant nor a common cousin.

After inverting the direction of the parent-child edges on our subgraph G_{bio}^{-1}, we run essentially a vanilla BFS algorithm on G_{bio}^{-1} to find the nearest common ancestor for each unioned pair of vertices in G. For each union edge $e \in E_u$ in the full genealogical network G, the search for a couple's nearest common ancestor proceeds backwards in time from both individuals in G_{bio}^{-1} simultaneously, adding one generation at a time to both trees, until the trees of ancestors either intersect or until the graph is exhausted and no additional ancestors can be added to either partner's tree. If the trees of ancestors intersect, then we can trace disjoint paths back to the nearest common ancestor, with one path commencing at the first partner and the second commencing at other partner. The total length ℓ of these paths is the couple's geodesic distance prior to union and this couple is said to have a *finite-distance union* of distance ℓ (see Figure 3.1). Note that this counting method places siblings at a distance of two, uncles and nephews at a distance of three, first cousins at a distance of four, first cousins once removed at a distance of five, and so forth. If no common vertex appears in both spouses' ancestry then the spouses share no common ancestor and this couple is said

to have an *infinite-distance union* (see Section 4.3).

While our algorithm ultimately identifies paths which travel forward in time from the nearest common ancestor, if present, to the given pair of vertices, our search algorithm constructs these paths in reverse. We remove outgoing edges and then convert incoming edges to outgoing edges when we search for paths from our pair of descendant vertices to their nearest common ancestor. Our search for nearest common ancestor does not permanently alter the orientation of any edges in the graph. The reorientation of some parent-child edges ensures that we search for ancestors rather than cousins or descendants and is temporary. Once a path is identified from a common ancestor to both objective vertices, all edges are restored to their original parent-to-child orientation (see Section 3.2).

The process of converting from a list of measured union distances to a probability distribution follows a slightly different process than is used to form $P_C(x)$ in the next section. When forming the PMF $P_C(x)$ of the number of children per household, if a certain value does not appear in the real world dataset—say for example that there are families in a dataset with three children and others with five children but no families have exactly four children—there is no consequence to our model's ability to correctly mimic the given real-world child distribution. We can randomly draw from a discrete probability distribution which has holes in its support and can correctly approximate the number of children in each household. Specifically, no households of exactly four children are necessary to have households of five or six children.

While our model is insensitive to any gaps in the child distribution's support, such gaps in the support of the distance to union distribution could cripple our model. For instance, one can imagine a scenario where no unions occur at distance seven in the real-world network, but a model grown by selectively forming unions between pairings of vertices could get stuck with a generation wherein all remaining possible pairings are at a distance of seven. In this scenario, such a modeled network would prematurely cease to grow. Furthermore, unions of lesser distances produce offspring which are at relatively greater distances. Vertices with a

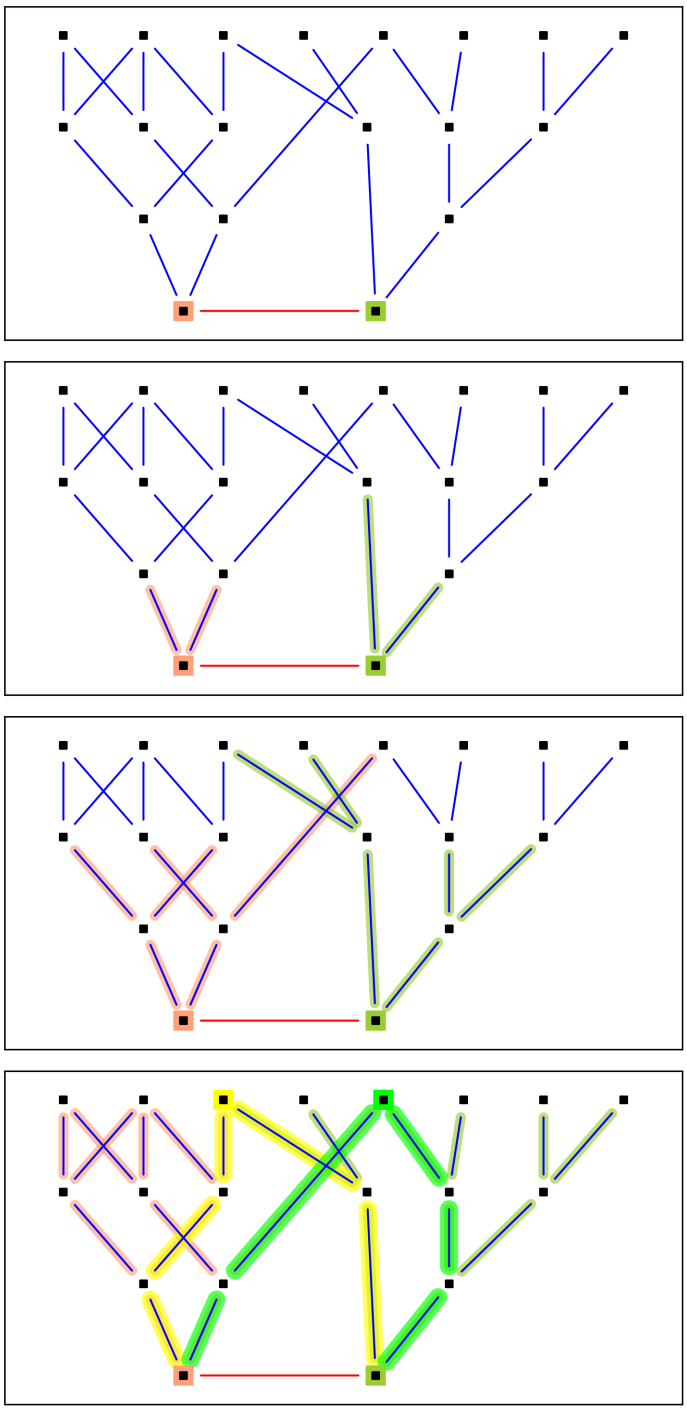

Figure 3.1: In this example, the pink and green vertices form a union (red edge). We search for their nearest common ancestor in the biological subgraph (along blue parent-child edges), building a tree of ancestors for both the pink and the green vertices one generation at a time until common ancestors are encountered. Both common ancestors (yellow and lime vertices) lie on paths of length five between the pink and green spouses, so we say that the pink and the green vertices have a finite-distance union at a distance of five, indicated here by the highlighted yellow and lime paths.

relative distance of four are necessary to later have vertices with a relative distance of five, six, or seven for example.

To remedy the challenges presented by possible gaps in the support of a real-world distance to union distribution, we form unions preferentially, selecting a candidate union at a distance outside of the support of real-world network's distance prior to union distribution $P_U(x)$ only when no other viable pairings remain (see Section 4.3).

Note that we do not adjust the support of $P_U(x)$. For example, in some island communities, marriages commonly occur at a distance of only two or three (i.e. siblings or an uncle/niece marriage, respectively) whereas in other cultures such close-relation unions are socially or legally forbidden and so do not occur. If such unions occur in the real-world network, then they will occur with a certain probability in an associated Target Model of that network.

3.3 Children Distribution

For each household in a list of union edges, we count the number of children. This list of counts is normalized to form a probability mass function (PMF) $P_C(x)$. More precisely, for a genealogical network with $u > 0$ union edges, we list the counts of children per household $(x_1, x_2, ..., x_u) \in \mathbb{N}_0^u$ where $x_i \geq 0$ represents the number of children in household i. This list of counts may include entries which are zeroes—i.e. it is possible that a given couple may have no children. No further edits or additions are made to the measured target distribution. If for example no family in the real-world community has four children, then the corresponding probability of a union in our model having four children is zero. No methods are employed to smooth the children per household distributions, to fill in gaps, or to otherwise coerce the measured data. We then divide the number of households with c children by the total number of households to form a PMF.

When we measure a real-world network for $P_C(x)$, we account for complete households only—those with children and two parents. Across our various datasets, single-parent house-

holds generally made up less than ten percent of the population, so we made the simplifying choice to exclude them from our measurements.

3.4 Size of Initial Generation

One of the parameters necessary for our model is not dictated by characteristics of the target real-world genealogical network. The number of individuals with which to begin our model in generation g_0 is purely a choice. While there are candidate vertices in a genealogical network for an initial generation, i.e. leaf vertices with children and no parents, there is little meaning in such a grouping of parentless leaf vertices. We can count how many such vertices are in a given genealogical network, but without extra information not conveyed by the graph itself we cannot know whether this group of parentless vertices coincides with a specific generation—i.e. we cannot know whether these leaf vertices are contemporaries of one another.

In the real world, generational divisions tend to accelerate and decelerate—e.g. a child born to thirty year old parents is farther away in time from their parents' ages than is a child born to teenage parents, but both relationships span only a single generation. The number of years between generations in an actual family varies widely and such differences aggregate together across spans of multiple generations. Instead of counting parentless leaf vertices and imposing the assumption that these leaf vertices somehow represent contemporary individuals, we propose searching for an initial number of individuals to place in g_0 using a bisection search method based on the rate of survival of the modeled genealogical network.

We define *survival* to mean that a modeled network has equaled or surpassed the size, measured in terms of the number of vertices, of the real-world genealogical network that we seek to mimic. If few unions form in a generation or if there is a high probability of having a less than two children per household, then it is likely that the subsequent generation will be smaller than the current generation. If a model network ever comes to a point where there are no available pairings with which to form unions or if ever all households in a generation

have no children before the modeled graph contains at least the target number of vertices n_{target}, then the model is said to have *died out*.

Our bisection search method proceeds as follows. For a given real-world genealogical network, we define our feasible set for starting size as the range between two and the number of individuals in the given real-world network $[a_0, b_0] = [2, n_{target}]$. This bisection search method begins at a random integer in the feasible set and then proceeds to construct modeled graphs beginning with that specified number of initial individuals, s_0.

If the modeled graphs die out more than a selected tolerance, then we begin constructing Target Model graphs with a larger initial population, taken as the midpoint of the previously-employed starting size s_0 and the upper bound of our feasible set n_{target}. Call this new starting size s_1. The range we are then searching within would be narrowed to the window between our first starting size and our original upper bound on our feasible set $[s_0, n_{target}]$, so that $s_1 = \lceil avg(s_0, n_{target}) \rceil$.

Similarly, if our instantiated Target Model graphs survive more often than our selected tolerance allows, then we begin constructing model graphs with a smaller initial population s_1, taken in this case as the midpoint of the original lower bound of our feasible set and the previously-employed starting size. In this case, the range within which we are searching would narrow to the interval between our original lower bound on the feasible set and our first starting size so that $[a_1, b_1] = [2, s_0]$ and $s_1 = \lceil avg(2, s_0) \rceil$.

The bisection search for an ideal starting size continues in this manner, narrowing the range between the previously-encountered upper or lower bounds and the midpoint between that bound and the previously-employed starting size until either the Target Model produces graphs which survive at the desired rate $r \in (0, 1)$ or until the upper and lower bounds are sequential integers (in which case, we take the larger as our ideal starting size).

In our numerical simulations, we sought for initial populations which were sufficiently large for the Target Model to survive $r = 95\%$ of the time, although this threshold could be adjusted up or down as desired. See Figure 3.2.

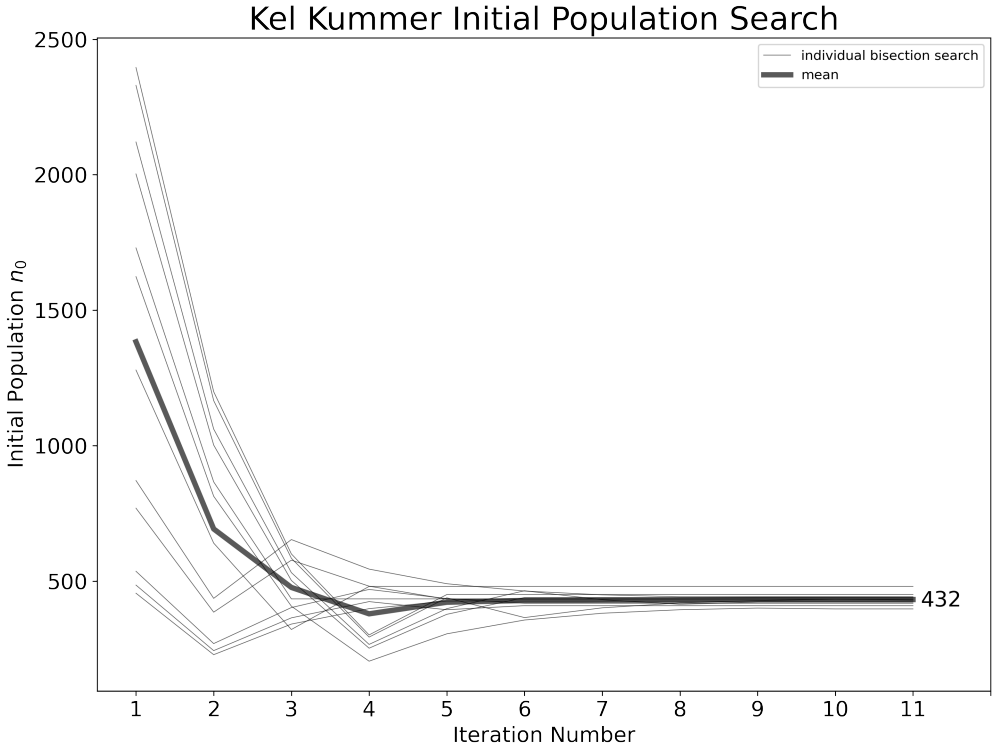

Figure 3.2: We conduct a bisection search for a reasonable initial population n_0, given a desired rate of survival. We chose to search for an initial population which would allow the Target Model to construct a graph containing at least as many vertices as the target network in 95% of instantiations. Each search begins at a random number between two vertices and $n_{target} = 2588$, the number of vertices in the real-world target network. If the model survives more often that the chosen 95% threshold, then a smaller initial population is selected for the next iteration; if the model survives less often than the desired threshold, then a larger initial population is take for the next iteration. We take our starting size n_0 as the mean of various independent bisection searches across the feasible set of initial populations $[2, n_{target}]$. For the Kel Kummer dataset, our bisection search method found a starting population of $n_0 = 432$.

3.5 Other Parameters

In addition to measuring and forming our two probability distributions $P_C(x)$ and $P_U(x)$, we note the size of the real-world network or number of vertices n_{target}, the fraction of finite-distance unions $p_{finite} \in [0, 1]$ which are those unions where the couples share a common ancestor, and the fraction of infinite-distance unions $p_\infty \in [0, 1]$ or those unions where the couples do not share a common ancestor. The probability that a vertex forms a union is p_{union} is taken as two times the number of unions in the given dataset divided by the number of vertices since we want to condition this probability on the number of vertices in the graph, as opposed to on the number of union edges. This union probability is the sum of the probability of finite-distance union and the probability of infinite-distance union

$$p_{union} = p_{finite} + p_\infty.$$

The complement of this sum $p_{single} = 1 - p_{union} \in [0, 1]$ is the probability that an individual remains single.

3.6 Running Example

As a concrete example, we introduce a genealogical dataset from the Menaka region of Mali in western Africa. This dataset centers on the Kel Kummer people, a more recent division of the traditionally nomadic Tuareg Iwellemedden people. While this particular dataset largely focuses on individuals between the mid-nineteenth to the mid-twentieth centuries, some of its ancestral lines can be traced back to the founding of the group in the seventeenth century [12, 13, 14].

This community's genealogy provides an example of the difficulty inherent to equating the passage of time with some fixed number of generations. Consider that the number of generations that separate the most recent individuals in the dataset from their most distant ancestors in the dataset varies widely from ancestral line to ancestral line. As an example, suppose that we adopted an average amount of time $T > 0$ between generations. If we

were to assume that mothers' average age at the birth of their child was $T = 25$, then we could estimate that about thirteen generations separate the most recent generation (meaning the most recent generation to be included in this curated genealogical network) of the Kel Kummer to their seventeenth-century foundations. In reality, however, some of the lines traversing that same time period contain twenty generations of ancestry [13]. No fixed amount of time can be used as a proxy for generational growth. While counting generations comes naturally in a theoretical model, where individuals may be added to a network at fixed, known intervals, real-world data has no convenient analog.

In terms of model parameters, the Kel Kummer genealogical dataset comprises some $n_{target} = 2588$ individuals and $m = 1011$ unions. It is nearly 56% male and 44% female and largely spans the century from the mid-1800s to the mid-1900s, though some lines can be traced as far back as the mid 1600s [12]. This dataset has a probability an infinite-distance union of $p_\infty = 18.5\%$ and a probability of a finite-distance union of $p_{finite} = 59.6\%$ for a combined total probability of union of $p_{union} = 78.1\%$. See Figure 3.3 for a visualization of the entire genealogical network and of its summary distributions and Table 3.2 for a summary of this dataset's statistics.

Chapter 4. The Target Model

As mentioned, the Target Model for genealogical networks accepts as input a real-world genealogical network $G = (V, E)$ and constructs artificial networks which approximate the structure of this network. This model provides a way to augment a specific genealogical network with many imitation networks which approximate its structure.

Our model forces generations to move jointly with time. We proceed iteratively forming unions among the current generation and introducing children vertices to form the next generation. In some ways this is in contrast to real-world families which have some blurring across generational lines. Consider for example an individual who is closer in age to their first

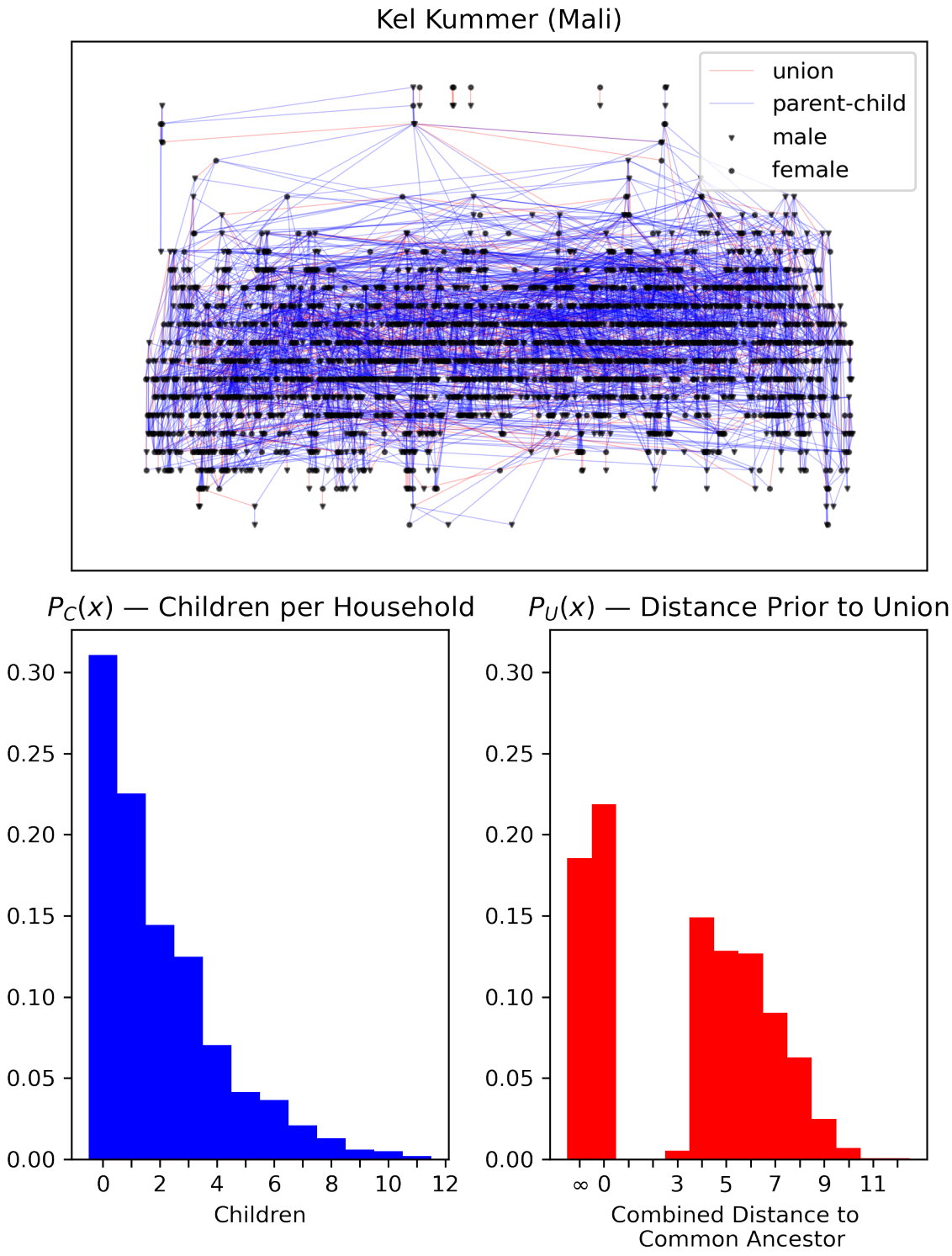

Figure 3.3: Top: An example of a genealogical network from the Kel Kummer people in Mali. Bottom Left: The distribution $P_C(x)$ of the number of children per union in the Kel Kummer genealogical network. Bottom Right: The distribution $P_U(x)$ of the distance prior to union in the Kel Kummer genealogical network. Infinite-distances indicate that a union formed between individuals who shared no direct line common ancestor. A distance of zero indicates the probability that an individual does not form a union.

Target Model Parameters		Kel Kummer
$P_C(x)$	PMF of Children per Union	See Figure 3.3
$P_U(x)$	PMF of Distance Prior to Union	See Figure 3.3
r	Chosen rate of survival for use with bisection search to find size of initial generation n_0	95%
n_0	Number of vertices for initial generation	432
n_{target}	Number of vertices in target genealogical network	2588
m	Number of union edges in target genealogical network	1011
p_∞	Probability of an infinite-distance union	18.5%
p_{finite}	Probability of a finite-distance union, the sum of the probabilities of forming a finite-distance union at each specific distance $d > 0$ in the support of $P_U(x)$	59.6%
p_{single}	Probability of not forming a union	21.9%

Table 3.2: Given a real-world genealogical network $G = (V, E)$, the parameters required for the Target Model are measured from G, with the exception of the size of the initial generation of vertices n_0, which we find using a bisection search to meet a user-specified survival rate r (see Section 3.4).

cousin's children than to their first cousins themselves. Or else consider that two individuals of about the same age might count three generations and four generations from themselves backwards respectively to find ancestors born one hundred years ago.

Our choice to force generational alignment in the Target Model restricts the pool of potential union partners for each vertex to those vertices which belong to the same, the immediately previous, or the immediately subsequent generation. Ultimately, our modeling choice that unions be restricted to neighboring generations is shown to be a reasonable one. By restricting our model's unions to pairings which span at most two immediately adjacent generations the Target Model can form unions at any distance from each union's nearest common ancestor. The Target Model also adds children vertices to all newly-formed unions at the same time. This choice is procedural and does not greatly deviate from real-world genealogical networks which tend to make no distinction in the network structure with edge weights or otherwise about how much time has elapsed between generation.

While counting generations is a temporal metric which is common in practice, it is always subjective in the real-world, providing relative distances between individuals in a family network. We simplify this temporal convention by forcing generations to move in lockstep. Our model adds one generation of individuals at a time and allows unions to form only between individuals in the same generation or between individuals in sequential generations. This simplification to keep generations sharply defined allows our algorithm to grow networks with a temporal sequence.

4.1 Algorithm

As introduced in Chapters 2 and 3, we begin with a given a genealogical network $G = (V, E)$ and its parameters (see Table 3.2). For a specified number of vertices $n_0 > 0$ the Target Model instantiates an initial generation g_0, comprising n_0 vertices and no edges. Each of the $n_0 \times n_0$ pairs of vertices in g_0 are assigned relative distances from one another via random draws from the distance to union distribution $P_U(x)$. Parent-child type edges and additional

vertices are added to the model to build these randomly drawn distances into the graph structure, connecting each pair of vertices in g_0 with a newly-added common ancestor with paths of the specified lengths. Beginning the modeled graph in this way immediately provides candidate unions at various distances from the support of $P_U(x)$ and allows the model to proceed generation by generation.

Once the initial generation has been constructed, the model proceeds to form unions from the possible pairings of vertices in the current generation as well as any still single vertices from the immediately previous generation (this set of still single vertices is empty when first forming unions of pairings in g_0). The total number of union edges to add is taken as the sum of the probability of an infinite-distance union p_∞ and the probability of a finite-distance union p_{finite} times half of the number of vertices n_0 in the initial generation g_0. We retain the other half of the initial generation to form unions with the next generation of vertices.

For the number of finite-union edges to be added, candidate pairs are first drawn at random from those possible pairings which would form at distances which appear in the support of our distance to union distribution $P_U(x)$, weighted by according to this distribution. If there are not enough candidate pairings, then the remaining number of finite-distance unions are drawn uniformly at random from whichever candidate unions are possible at the distance nearest to the support of $P_U(x)$. The number of infinite-distance union edges are each connected to a new vertex, i.e. an *immigrant*, or a vertex which has no incoming parent-child edges, and a yet single vertex in g_0.

An independent random draw from the distribution of children per union is made for each union edge formed. We connect these new children vertices to both of their parent vertices via directed parent-child type edge (directed from parent to child). Together, these children make up the first generation g_1. The algorithm continues in like manner, iteratively forming unions and adding children until either no children are added, no unions are possible or until the total number of vertices in generations $g_0, g_1, ..., g_i$ equals or surpasses the number of vertices n_{target} in G.

The algorithm for the Target Model may be summarized as given below:

- Create the initial generation g_0 of n_0 individuals, forming unions between them (and possibly some immigrants) using $P_U(x)$.
- Add children to each union in g_0 using $P_C(x)$ to create our next generation g_1.
- Add infinite-distance individuals to g_1 and form unions in g_1 with probabilities p_∞, p_{finite}.
- Form mixed generational unions between still single individuals in g_1 and in the immediately-previous generation g_0, again using the probabilities in $P_U(x)$.
- Repeat for each subsequent generation g_k until the size of the network exceeds n_{target}.

4.2 Initialization and Number of Generations

For our initial generation of individuals g_0, consisting of n_0 vertices and no edges, there are $n_0 \times n_0$ possible pairings of individuals. For each possible pairing in this initial generation we draw a value from $P_U(x)$. These distances are then imposed on the graph. That is, between each pairing of vertices in g_0, we construct a path of the specified length of parent-child edges (introducing the corresponding number of new vertices to the graph). These new paths do not intersect; each pair of vertices in g_0 has its own distinct common ancestor and necessary scaffolding (i.e. intermediate vertices and edges on the path to the common ancestor from each vertex in the pair) created. These paths provide the structure necessary to create the specified relationships between vertices in our initial generation.

For each generation $k = 0, 1, 2, ..., L$, we define a matrix $D^k = [d_{ij}] \in \mathbb{N}^{n \times n}$ to track the distances between all pairs of eligible individuals (those still single individuals in the current and immediately previous generation) where $d_{ij}^k \in \mathbb{N}_0$ is the distance from individual i to individual j. Note that D^k is symmetric so that $d_{ij}^k = d_{ji}^k$ and $d_{ij} \equiv 0 \iff i \equiv j$. (In our accompanying code, infinite distance marriages are encoded with a distance of -1.) For our

initial generation, we store the distances between each pair of vertices in a symmetric matrix $D_0 \in \mathbb{N}_0^{n_0 \times n_0}$.

After the initial generation is instantiated, we build out each successive generation by iteratively forming households, generation by generation until our constructed graph surpasses n_{target}, the number of vertices in G, so that our artificial networks approximately match the size of our real-world target networks. Specifically, our model operates one generation at a time until the total number of vertices from the initial generation g_0 to the final generation g_L surpasses our target size, i.e. $\sum_{k=0}^{L} |g_k| \geq n_{target}$.

4.3 Form Unions

The Target Model does not track the sex of each vertex. While the real-world networks have a certain division of men and women, the Target Model treats all vertices equally and thus assumes that the sex of each individual was whatever sex was needed to form each union at the desired distances.

Our model forms three types of unions: infinite-distance unions, intergenerational finite-distance unions, and intragenerational finite-distance unions. Not every union in a real-world genealogical network $G = (V, E)$ will share common ancestors. In some unions one partner may have immigrated to or emigrated from their community. Individuals in G form unions with immigrants to their community—defined as persons with whom they share no common ancestry—at average rate p_∞. Likewise, our model introduces new vertices so that a random $p_\infty \in [0, 1]$ fraction of the individuals not in a union in the current generation form unions with persons who are an infinite distance away from them—i.e. the pair of vertices has no common ancestry in the model.

The remaining unpaired vertices are divided into two camps. Half of these vertices are designated to form unions with other vertices from the following generation while half are designated to form unions with other vertices from among the current generation and the still eligible vertices from the previous generation. Among the vertices to be paired off this

generation and the designated vertices from the previous generation, we note how closely each possible pairing is related to one another in D^k, then randomly draw marriages, weighted by the probability of a marriage of each respective distance occurring (see Section 4.3.2). We purposefully prevent pairings where both spouses come from the previous generation of vertices. By favoring these intragenerational pairings we preserve the possibility of odd-valued distances to common ancestry.

4.3.1 Mixed Generation Unions Can Preserve All Distances. Consider that our generation-by-generation growth pattern increments the length of existing paths by two edges each time that children are added to a generation's families—that is the if two households have a shortest biological ancestral path between them that is of length d, then the children of these two households will have a such a path that is length $d + 2$ between them.

Specifically, as our model grows, relative biological distances cannot decrease. In order to have candidate pairings at a greater distance, we must have formed pairings at lesser distances in a previous generation. Considering our restriction that unions form only among the same generation or between adjacent generations, there must be individuals which are at a distance of $d - 1$ or $d - 2$ in the immediately preceding generation if in the current generation there will be pairings possible with which to form a union of distance d. Because later generations rely on the prior presence of lower-distance relationships in order to form higher-distance unions, we give each generation the opportunity to form both even- and odd-distance unions.

If we were to restrict the unions we form to those pairings where both partners come from the same generation, then odd-valued distances cannot be introduced in subsequent generations, creating an even-dominated distance to union distribution in our modeled graph. This even-dominated distance to union distribution occurs because existing paths are always lengthened by two edges at a time in our model, so as the model grows, any pairing which traces its nearest common ancestor to a vertex in any generation $g_0, g_1, ..., g_{L-1}$ (i.e. any pairing whose nearest common ancestor is not among the loops constructed to impose

distances on the individuals in our initial generation g_0) will have an even-distance prior to union. In order to preserve the possibility of forming unions at both even and odd distances, we force half of each generation to wait to form unions until after the next generation of individuals become eligible.

4.3.2 Reemphaisize Distances. Beginning with generation two, we bias the probability distribution with which we select which unions to form to favor those distances which have occurred less frequently than the target distribution would indicate. This update scheme affects the proportions of individuals who will form finite-distance unions, will form infinite-distance unions, and will not form a union (i.e. will remain single).

- Take the list of union distances currently in the model, i.e. all those unions created from generation g_0 to the previous generation g_{k-1}, and normalize the counts into a probability distribution, $P_U^{k-1}(x)$.
- Subtract $P_U^{k-1}(x)$ from the original target distance to union distribution: $\Delta_U^k = P_U(x) - P_U^{k-1}(x)$.
- Where $\Delta_U^k < 0$, set $\Delta_U^k = 0$.
- Normalize Δ_U^k to form the PMF $P_U^k(x)$ from which union pairings will be drawn in our model at generation k.

The resulting probability distribution emphasizes those distances which have not, as of the previous generation, occurred as frequently as they ought to have occurred and corrects for over- or under-occurrences of infinite distance unions, finite distance unions, and the probability of remaining single.

4.4 Add Children

After forming marriages and adding marriage edges to the graph, the next generation is populated. For each union edge formed in our model, we make an independent random draw

from G's distribution of children per household $P_C(x)$. The sum of these random draws is the total number of children to add to the graph and these new vertices constitute our next generation g_k. We introduce the corresponding number of vertices and add parent-child edges between each child and both of their parent vertices for each household. There is a possibility that a household will have no children.

The model then prepares to execute afresh—adding immigrants, forming likely unions, and introducing children—to this newly introduced generation of individuals. This preparation includes forming a new matrix D^k of distances between each possible pairing of vertices in the upcoming union-forming step. This matrix D^k is square and symmetric, tracing the distance to common ancestor between each pair of vertices in the current generation g_k as well those vertices from the previous generation g_{k-1} which were not given the opportunity to form unions previously.

4.5 Stopping Criteria

The model runs generation by generation forming unions and introducing children to the successive generation until the total number of vertices in the modeled graph's generations $g_0, g_1, ..., g_L$ (i.e. excluding the vertices preceding the initial generation g_0) equals or surpasses the real-world network's size n_{target}, measured in number of vertices. If ever a generation occurs where no new unions can form or where no additional children are introduced to the next generation, then the model stops. (Our accompanying code base also optionally allows the user to specify a fixed number of generations for the model to construct, but we feel that this stopping criteria fails to account for the real-world complexity that generational lines blur across even closely related families.)

4.6 Target Model Outline

The steps used to generate a genealogical model using the Target Model algorithm are explained in detail below:

Stage 1: Initialization:

(a) Set k=0.

(b) Begin with $n_0 > 0$ vertices, add these n_0 vertices to the empty graph M. These n_0 vertices comprise our first generation g_0.

(c) For each of the $n_0 \times n_0$ pairs of individuals in g_0, draw randomly from $P_U(x)$. Record these distances in a symmetric matrix $D^0 = [d_{ij}^0]$ where $d_{ij}^0 \sim P_U(x)$.

(d) Build out the graph structure to represent the distances in D^0. For each pair of vertices (i, j) in g_0, add a path of length d_{ij}^0 between vertices i and j.

Stage 2: Growth: While $length(g_k) > 1$ and while there are fewer vertices from the initial generation to the previous generation inclusive than the total number of vertices in the target network G (i.e. while $\sum_{k=0}^{L} n_k \leq n_{target}$):

(a) If $k > 1$, form $P_U^k(x)$ to increase the probability of underrepresented distances in the support of the distance to union distribution $P_U(x)$.

(b) Form unions.

Step 1: Multiply the number of individuals in the current generation g_k by the probability of forming an infinite- and of forming a finite- distance marriage p_∞ and p_{finite}, respectively, to find the number of infinite-distance and finite-distance unions to create. Round to the nearest integer in each case.

Step 2: Randomly divide the current generation into two sets: first, those individuals who will attempt to form a union with another individual from the current generation or with a still-single individual from the previous generation and

second, those who will not attempt to form a union until the next generation—i.e. this second category will remain single this time through the while loop, but may pair up next time we execute the while loop.

Step 3: Form a list of all possible pairings consisting of pairs where at least one partner comes from the list of those individuals who can form a union this generation (the second partner may come from the list of yet single individuals who did not form unions in the previous generation). Record the relative distances prior to union for each candidate pairing listed.

Step 4: Divide this list of all possible pairings into two sets: first, a list of preferred unions, consisting of those pairings which would occur at a distance which appears with non-zero probability in the support of $P_U^k(x)$ and second, a list of other unions which would occur at a distance other than those found with non-zero probability in the support of $P_U^k(x)$.

Step 5: While preferred unions remain and fewer finite-distance unions have formed than calculated, randomly select a finite-distance pairing with probabilities given by $P_U^k(x)$. For each pairing selected, add a union edge to the graph and update the list of available preferred pairings by removing all other candidate pairings which contained either partner now connected by the new union edge.

Step 6: While other unions remain and fewer finite-distance unions have formed than calculated, identify which possible finite-distance pairings would occur at a distance closest to the support of our original target distance to union distribution $P_U(x)$. Of the available pairings at this closest-to-correct distance, select one uniformly at random, add a union edge to the graph, and update the list of available other pairings.

Step 7: Update the number of infinite-distance unions to form. Take it as the minimum of the number of vertices remaining to be paired off this generation and the number of infinite-distance unions to form that was calculated in Step 1.

Step 8: Uniformly at random select individuals from the still-single individuals in the current generation g_k and in the previous generation g_{k-1} to marry at an infinite-distance—i.e. to marry those with whom they share no common ancestor. Add the corresponding number of new vertices to the graph and add the union edges connecting the selected individuals to the newly introduced vertices.

(c) Add children. For each union edge added in Steps 5, 6, and 8, draw randomly from $P_C(x)$ and introduce the corresponding of new vertices to the graph. Add parent-child type edges between the new children vertices and each of their parent vertices. Note that each child vertex will have two incoming parent-child type edges, one from each of the parent vertices in the union.

(d) Update the distance to union matrix to find D^k. This matrix tracks the distance between each possible pairing of the children introduced to the graph this generation and each of those individuals from the previous generation which were selected not to form unions until the next generation.

(e) Increment k.

Chapter 5. Variant Algorithm

While our algorithm (see 4.1) produces graphs which reasonably approximate the target distance to union distribution, it comes at a cost. The choice to actually build out paths to the initial generation's nearest common ancestors adds a substantial number of individuals to the graph, particularly if a larger initial population n_0 is selected. Consider for example, a target network which requires an initial population of one hundred individuals for $r = 95\%$ of model instantiations to survive (see Section 3.4). If the expected value of the target network's distance prior to union distribution is 10, then for the 1000 possible pairings in the initial generation our model would, on average, add nine new vertices and ten new parent-child

type edges for each pairing to the graph (see steps Stage 1 c, Stage 1 d in Section 4.6). These added setup vertices and edges create a model which can immediately begin to form unions for candidate couples at all distances, but at the cost of ballooning the overall size of the modeled graph by many thousands of vertices and edges. This ballooned size can present computational issues—for example Zachary Boyd et al. present a persistent homology based methodology for comparing genealogical networks with social networks [15]. They argue that persistence curves may be used to distinguish between social networks and genealogical networks, but they warn that their algorithm for calculating the persistent homology has a spatial and a temporal complexity of $O((n+m)^3)$ where n and m are the number of vertices and the number of edges in the graph, respectively. Such a complex computation would prove prohibitive for many of the graphs produced by our algorithm 4.1. As such, we propose an slight variant to our algorithm which still produces a single weakly connected component, but which drastically reduces the number of additional vertices and edges needed to connect the graph into a single component.

5.1 Variant Model Initialization

In this variant algorithm, we largely follow the same procedure as before, but with some notable differences. Rather than connecting every possible pairing of individuals in the initial generation with paths from their common ancestor, we instead treat every pairing in the initial generation as sharing no common ancestor. With every vertex in the initial generation at an infinite distance from every other vertex in the initial generation, only infinite-distance unions are possible during the initial generation. The variant algorithm still proceeds generation by generation iteratively forming unions between pairings in the current generation and adjacent generations, but the choice to begin with vertices which share no common ancestry creates the need for a burn-in period before finite-distance unions are possible. A number of generations of unions and children must be added to the graph before any finite-distance candidates (i.e. pairs of vertices which share a common ancestor)

exist.

5.2 Variant Model Infinite-Distance Unions

In the variant algorithm, we allow infinite-distance unions to form between two vertices which are already in the graph. Any two vertices which are not a part of to the same weakly connected component share no common ancestors. The converse however is not true—immigrant vertices connected by a union edge belong to the same component but share no common ancestry with any other vertices in the component. Unlike our primary algorithm (Chapter 4), not every infinite-distance union requires a new vertex (i.e. an immigrant) to be introduced to the graph, rather our variant algorithm prioritizes forming infinite-distance unions between vertices which are both already in the graph. Whenever an infinite-distance union forms between two vertices which have already been added to the graph (i.e. each partner vertex has two parent vertices in the previous generation), then the union connects two weakly connected components of the graph into a single weakly connected component. If no such infinite-distance candidate pairing exists within the graph, then the variant algorithm follows the original algorithm's format and uniformly selects partners from the current generation to be paired up with newly-introduce immigrant vertices.

5.3 Connecting Variant Model Components

Even though we connect components in our modeled graph by forming infinite distance unions with two already-introduced vertices whenever possible, the variant algorithm typically results in a graph with many disjoint components. Having multiple disjoint components is not necessarily unrealistic—one can imagine a community with a few predominant families to which most but not all individuals in the community are related—but it is problematic for persistent homology calculations on the graph, etc. For instance, dimension one persistent homology calculations only consider a single connected component of the graph. In

the interest of pursuing a model which will produce graphs for which persistent homology calculations are both feasible and appropriate, we employ the following methodology to form a single connected component in graphs produced by our variant model.

We aim to form a single weakly connected component in the graph produced by the Variant Target Model. We do so by connecting components together via parent-less vertices in each component. These parent-less vertices are immigrants to the family, meaning that they have no incoming parent-child type edges. To form one component of two disjoint components, we identify a vertex in each of the two components which have fewer than two incoming parent-child type edges. The identified pair of vertices, one in each of the previously unconnected components, remain unioned to partners in their original components. We choose to enforce that immigrant vertices must pertain to neighboring or to the same generation. This choice ensures that realistic temporal connections will be added to the graph—we will not accidentally say that a vertex in the initial generation g_0 is siblings with some immigrant vertex that was introduced in the twentieth generation of the model, for instance. We begin our search for candidate vertices to connect via a common ancestor in the latest generation added to the graph and proceed to search backwards in time until a suitable pairing is identified. In this way any new common ancestry introduced to the graph will overlap temporally with the other structure of the modeled graph to the extent possible.

With a candidate pairing identified, we then draw from the target distance to union distribution $P_U(x)$. The drawn value indicates how closely the two vertices are to be related. As in the original Target Model instantiating (see Section 4.2), we introduce a new vertex as the most recent common ancestor of the identified pair of vertices and construct paths of parent-child type edges from this common ancestor to each of the identified vertices, introducing additional new vertices as needed. As before, the total length of these two paths is the distance to common ancestor for this pair of vertices. No union edges are introduced. We continue this joining up process until the modeled graph contains a single weakly connected component.

5.3.1 Connecting Variant Model Components Outline. After completing Stage 1 and Stage 2 from the algorithm in Section 4.6 with the modifications outlined in Sections 5.1 and 5.2, complete the following:

Stage 3 Connecting Disjoint Components: While more than one component in the graph M produced by the Target Model:

(a) Take two components of the graph M. For each component, make a list of vertices with fewer than two incoming parent-child edges, noting the generation number in which each vertex was added to the graph.

(b) Sort the lists of available vertices in each component by the generation in which they were introduced to the graph.

(c) Beginning at the latest generation in one of the components' lists of available vertices, search the other component's list for available vertices belonging to the same generation or in the next or previous generation. If such a vertex exists, then uniformly at random select one of the candidate vertices from the first component's set of available vertices in the latest generation and likewise select one of the candidate vertices from the second component's set of available vertices in the identified generation. These randomly selected candidate vertices form the pair of immigrants to which we will introduce a common ancestor. If the second component does not contain any candidate immigrant vertices in the same generation or in the previous or in the next generation, then decrement the search generation and repeat the search in both components, with the search window shifted backward in time.

(d) Draw randomly from $P_U(x)$. This will be the distance between the pair of candidate vertices identified.

(e) Build out the graph structure to represent the distance between the pair of vertices. Introduce a new vertex as the common ancestor and other new vertices as

necessary to build a path of parent-child type edges from the common ancestor to both of the vertices in the pair. These new paths will form a single weakly connected component out of the two previously disjoint components.

Chapter 6. Results

We summarize the effectiveness of the Target Model (Algorithm 4.6) and its variant (see Section 5.3.1). We also present results measured on subgraphs of the networks produced by the Target Model. These subgraphs follow the algorithm outlined in Section 4.6 and then omit all vertices and edges preceding the initial generation g_0. These auxiliary ancestry paths connect pairs of vertices in g_0 to a common ancestor. Their presence in the graph allows the Target Model to immediately begin forming unions of all distances, but the additional structure distorts the probability distribution that a random vertex in the graph will form a union at specific distances. The vast number of auxiliary vertices simply overwhelms the relatively small number of vertices contained in generations $g_0, g_1, ..., g_L$.

The original Target Model does very well at approximating the number of union edges at each distance but does poorly at capturing the probability distribution $P_U(x)$, because of the large number of pre-generation 0 vertices. If we run the Target Model and then examine the subgraph containing only those vertices and their out-going edges in generations $g_0, g_1, ..., g_L$, then the modeled network necessarily shows an increase in infinite-distance unions because the initial generation no longer share any common ancestry. These subgraphs capture $P_U(x)$ more closely than the full Target Model networks because there are relatively fewer vertices in the graph on which the probability of a vertex participating in a union depends. The variant algorithm grows without imposing common ancestry on the initial generation. As a result, the Variant Target Model shows much greater variance in survival rates, tends to require much larger initial populations, and produces networks with few generations that are much larger than the original Target Model produces.

6.1 Target Model—Results With Auxiliary Ancestry

The Target Models initializes with a fully connected initial generation. Every possible paring in the initial generation g_0 shares a common ancestor. These common ancestors do not belong to the initial generation. Together with the parent-child type edges and additional intermediate vertices comprising the paths from common ancestor to individuals in g_0, these common ancestors introduced to the modeled network are *auxiliary ancestry*. They are necessary to provide candidate union pairings at finite distances, but are not counted toward the stopping criteria of having at least n_{target} vertices for the Target Model. The choice to exclude the auxiliary ancestry from the stopping criteria largely comes from the large number of vertices introduced in these pre-g_0 lines. For example, the Kel Kummer dataset requires a starting size of $n_0 = 432$ individuals (about 19% of the Kel Kummer network's total size). Connecting every possible pair to a common ancestor at a distance drawn from $P_U(x)$ results in an additional 450,000 vertices. The original Kel Kummer network comprises only $n_{target} = 2588$ individuals.

As seen in Figure 6.1, there are advantages and disadvantages to the modeling choice to fully connect g_0 with so many auxiliary ancestral lines. Including the auxiliary lines allows finite-distance unions to form immediately without any burn-in generations. Because the auxiliary lines contain no union-type edges and because these lines support the immediate formation of finite distance unions, the distribution of distance to common ancestor between unioned partners is well captured. Throughout the target model, this distribution is related back to $P_U(x)$ whenever we form $P_U^k(x)$ when forming unions in the kth generation g_k (see Section 4.3.2). $P_U(x)$ is the probability that a vertex will participate in a union at at specific distance. As measured in a real-world genealogical network, $P_U(x)$ is conditioned on being unioned. The Target Model transforms these probabilities to give the probability that a vertex will participate in a union at a specific distance, without conditioning for unioned vertices only. We provide two views of the distance to union distributions in the Target Model with auxiliary ancestry. First, we examine the model's distance to union distribution

when conditioned on a vertex's belonging to a union. Second, we examine the same without conditioning on a a vertex's belonging to a union. In the latter, we present probabilities which account for the entire network structure, including the auxiliary ancestral lines. This inclusion overwhelms the small proportion of unioned vertices, such that only the probability that vertex remains single is significant.

All variants of the Target Model presented, including the Target Model with auxiliary ancestry share the same distribution of children per household. When calculating a modeled network's distribution of the number of children per household, we condition on union edges. Only a pair of vertices joined by a union edge can have children in the Target Model, so the number of households in the modeled network is determined by the number of union edges in the network. The auxiliary ancestry preceding the initial generation g_0 does not contain any additional union edges, so the Target Model with auxiliary ancestry and the Target Model omitting auxiliary ancestry after model growth both share the same set of union edges. That we approximately recapture $P_C(x)$ through independent identically distributed random draws is unsurprising, but does support that the Target Model mimics the original target network's local structure (see Figure 6.2)

In order to compare the global structure of Target-Model-produced graphs against that of their real-world counterparts, we examine the cycle structure of each. The Target Model with auxiliary ancestry produces networks with cycle bases which approximate the cycle basis of the target real-world network. Given a graph G with a spanning tree s, a *cycle basis* or set of *fundamental cycles* is a set of cycles c_i where each cycle contains "exactly one non-tree edge each." These fundamental cycles "can be combined into new cycles by adding their edges modulo two, i.e. if the same edge is covered an odd number of times it is kept and otherwise discarded" [16]. When compared to a configuration model executed on the Target Model's degree distribution, a random graph in the same class as the network produced by the Target Model, the Target Model produces networks whose cycle bases are closer to that of the target real-world network, although still with more cycle basis elements than are present

in the real-world network's cycle basis. Over 1000 instantiations on the Kel Kummer dataset, the Target Model produces networks whose distribution of lengths of cycles in the their cycle bases more nearly approximates the Kel Kummer network's distribution of length of cycles in its cycle basis (see Figure 6.3). Across many networks, the Target Model demonstrates the same phenomena. Figure 6.4 shows that the Target Model produces networks with better than random cycle distributions, even across diverse real-world datasets.

6.2 Reduced Target Model—Results Omitting Auxiliary Ancestry After Model Growth

If we run the regular Target Model algorithm (see Algorithm 4.6)—introducing an initial generation of individuals g_0 and then fully connect them with auxiliary ancestral lines, giving each pair of individuals in g_0 a common ancestor, then grow the network until the total number of individuals in generations $g_0, g_1, ..., g_L$ (excluding all auxiliary ancestry connecting pairs in g_0) exceeds the target size n_{target}—and then examine the subgraph of the resulting network comprising only $g_0, g_1, ..., g_L$, we produce a network which approximates the target network's size n_{target}. This reduction in size does not affect the distribution of children per household (see Figure 6.2).

Removing the auxiliary ancestry reconciles the differences observed in the distance to union distributions presented in Figure 6.1. Observe in Figure 6.5 that the distance to union distribution conditioned on a vertex belonging to a union (Top Left) agrees with the probability distribution's shape when not conditioned on being unioned (Bottom Left). While omitting auxiliary ancestry from the Target Model after model growth creates agreement between both notions of distance to union distribution, the modeled network has too large of a proportion of infinite-distance unions. When we remove the common ancestry connecting each pair of vertices in g_0, the unions in g_0 convert to infinite distance unions.

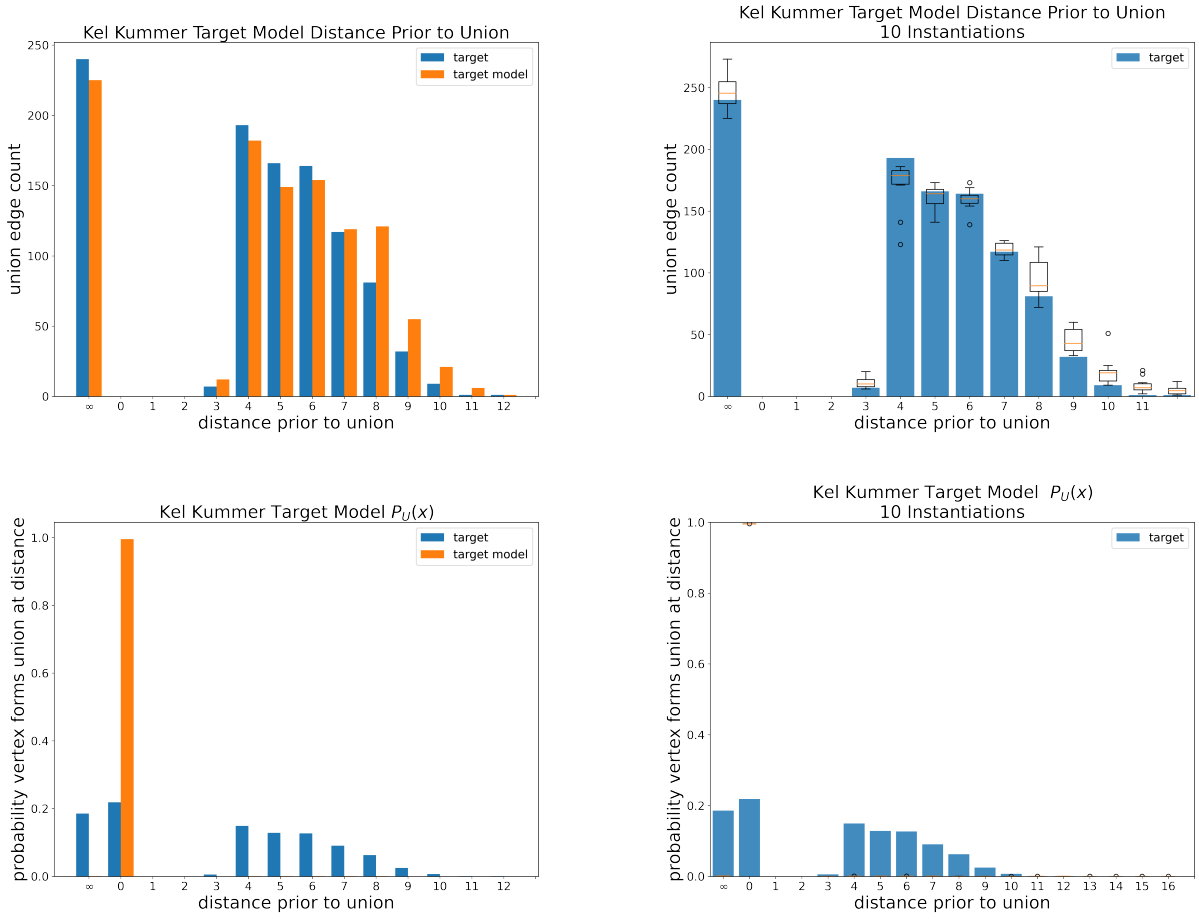

Figure 6.1: The Target Model as presented in Section 4.6 results in the following comparisons for the Kel Kummer dataset. The Target Model captures the distribution of distance to common ancestor, with reasonable consistency. These distributions show the probability that a randomly selected union edge in the graph will be a certain distance from their partner (Top Row). The Kel Kummer dataset requires an initial population g_0 of 432 individuals. The Target Model introduces approximately 450,000 auxiliary vertices to impose a relative distance prior to union for each pairing in this initial generation, resulting in a near-zero probability that a randomly-selected vertex in the graph will be partner to a union. By including the auxiliary ancestry which connects all pairings in the initial generation, the resulting probability distribution in the model $P_U^L(x)$ (Bottom Row, orange) hardly resembles that of the target real-world network $P_U(x)$ (Bottom Row, blue). Contrast these results with those presented in Figure 6.5, where our calculations of distance prior to union exclude the auxiliary structure prior to g_0.

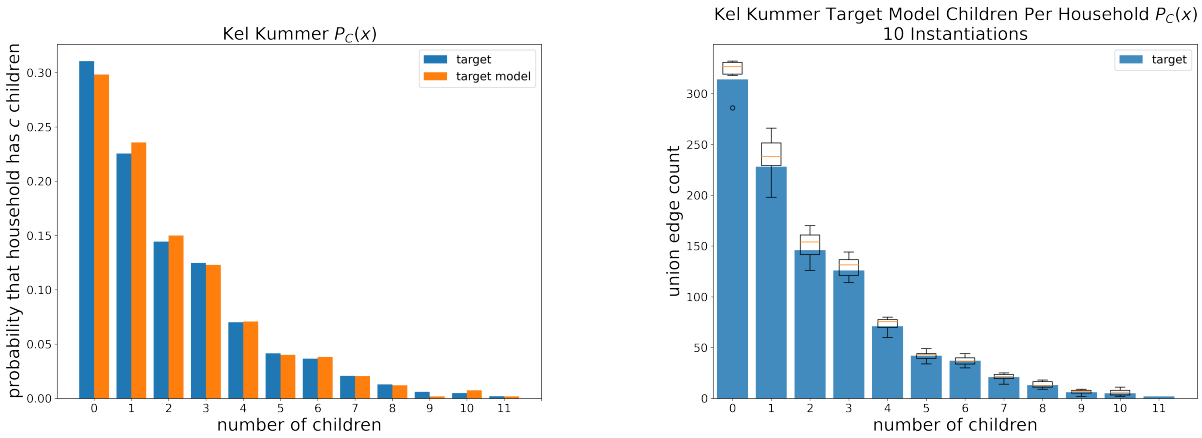

Figure 6.2: The Target Model as presented in Section 4.6 results in the following comparisons for the Kel Kummer dataset. Left: Each draw from the target network's distribution of children per household $P_C(x)$ is independent and identically distributed. This distribution is calculated per union edge, so that the auxiliary structure which may precedes the initial generation g_0 has no effect on the resulting children per household calculations. Right: The real-world target network's distribution of children per household (blue) is compared against the model's distribution of children per household for ten instantiations. That we approximately recapture this distribution from independent random draws is unsurprising, but necessary to show that our modeled networks approximately capture the real-world network's local structure.

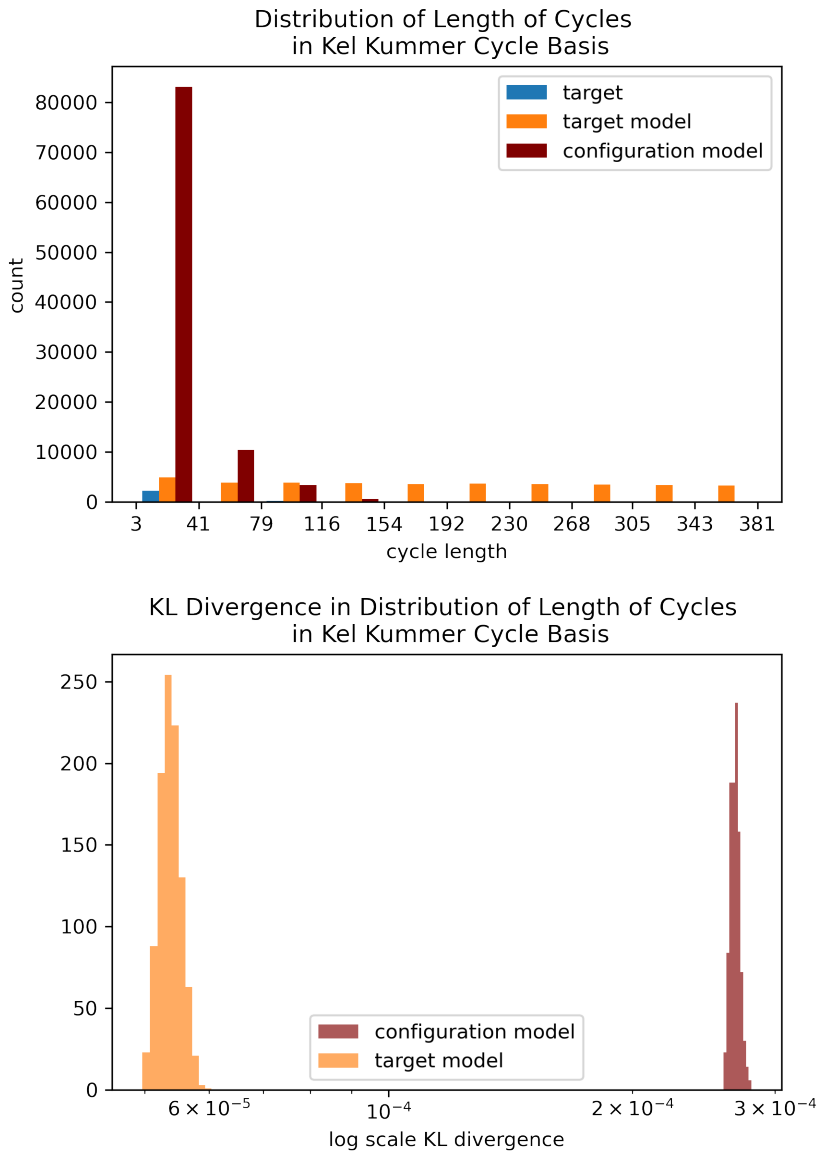

Figure 6.3: We run the Target Model as presented in Section 4.6, retaining the auxiliary ancestry preceding the initial generation g_0. Top: For a single instantiation of the Target Model using the Kel Kummer dataset, we compare the lengths of the fundamental cycles of the undirected real-world target network (blue) against that of an undirected Target Model (orange). We further compare the Target Model against a configuration model, created using the Target Model's degree distribution. Notice that the Target Model's cycle basis more nearly approximates the distribution of lengths of cycle basis elements in the real-world target network. Bottom: For 1000 instantiations of the Target Model using the Kel Kummer dataset, we form a configuration model based on the Target Model's degree distribution. We then measure the lengths of elements in the cycle basis for the Target Model and for configuration model, then we measure the KL divergence between the Target Model's distribution of lengths of cycle basis elements against that of the target network. We do the same for the configuration networks. These histograms show that the Target Model's distribution of cycle lengths more nearly approximates the distribution of cycle lengths in the target network. Contrast these results with those presented in Figure 6.6, where we exclude auxiliary structure before calculating cycle bases.

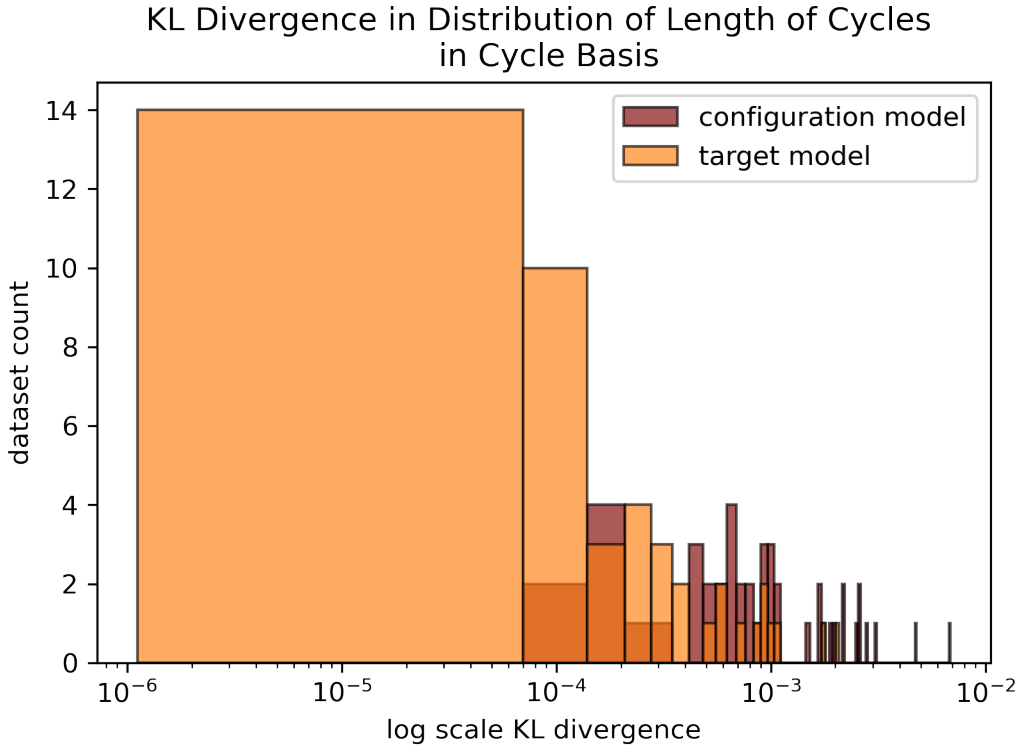

Figure 6.4: For each of the real-world target networks (see Appendix A), we instantiate 1000 Target Models, retaining the auxiliary ancestral lines preceding the initial generation g_0. For each of the Target Models, we instantiate a configuration model. For each of the 1000 Target Models and for each of the 1000 configuration models, we find a cycle basis, and measure the length of each element in the basis to form a distribution of lengths of cycle basis elements. We then measure the KL divergence between each Target Model graph's distribution of lengths of cycle basis elements and that of the target real-world network. We take the median KL divergence score from the set of 1000. We treat the configuration models similarly, finding the KL divergence between the target network's distribution of lengths of cycle basis elements and that of each of the 1000 configuration models, recording the median performance. A smaller KL divergence indicates better agreement between distributions and we note that almost universally the Target Model produces networks whose cycle bases are more similar to the cycle bases of the corresponding real-world genealogical networks.

In Figure 6.6, the network produced by omitting auxiliary ancestral lines from a Target Model on the Kel Kummer dataset is shown to largely agree with the cycle structure of the real Kel Kummer network. The Target Model significantly outperforms a configuration model based on the Target Model's degree distribution (Top). Across our available datasets, omitting auxiliary ancestral lines from the networks produced by the Target Model produce better cycle structure agreement with their real-world counter parts than does a purely random graph. Across 1000 instantiations of a Kel-Kummer-based Target Model and corresponding configuration models, the cycle basis in the Target Model networks more closely agrees with that of the real Kel Kummer dataset, see Figure 6.6 (Bottom). This agreement is demonstrated by smaller KL divergence between the distribution of the length of cycle basis elements in Target Model produced networks and that of the real Kel Kummer dataset than is observed between the distribution of the length of cycle basis elements in the configuration model produced networks and that of the real Kel Kummer dataset.

This trend appears across datasets. Target Model networks consistently have fundamental cycles which more closely mimic the lengths of their real-world counterparts than do similar configuration model produced graphs. Again, for each dataset, we run 1000 Target Models, omitting the auxiliary ancestry post network growth. Then for each of these 1000 Target Models we instantiate a configuration model based on the Target-Model-produced network's degree distribution. For each of the 1000 Target Models and for each of the corresponding 1000 configuration models we find a basis of the fundamental cycles in each graph. We then find the distribution of cycle lengths in each of these fundamental cycle bases and then take the KL divergence between the distributions of cycle lengths in the Target Model and in the real-world network and likewise between that of the configuration model and the real-world network. We then record the median performance. These median KL divergences are show in Figure 6.7.

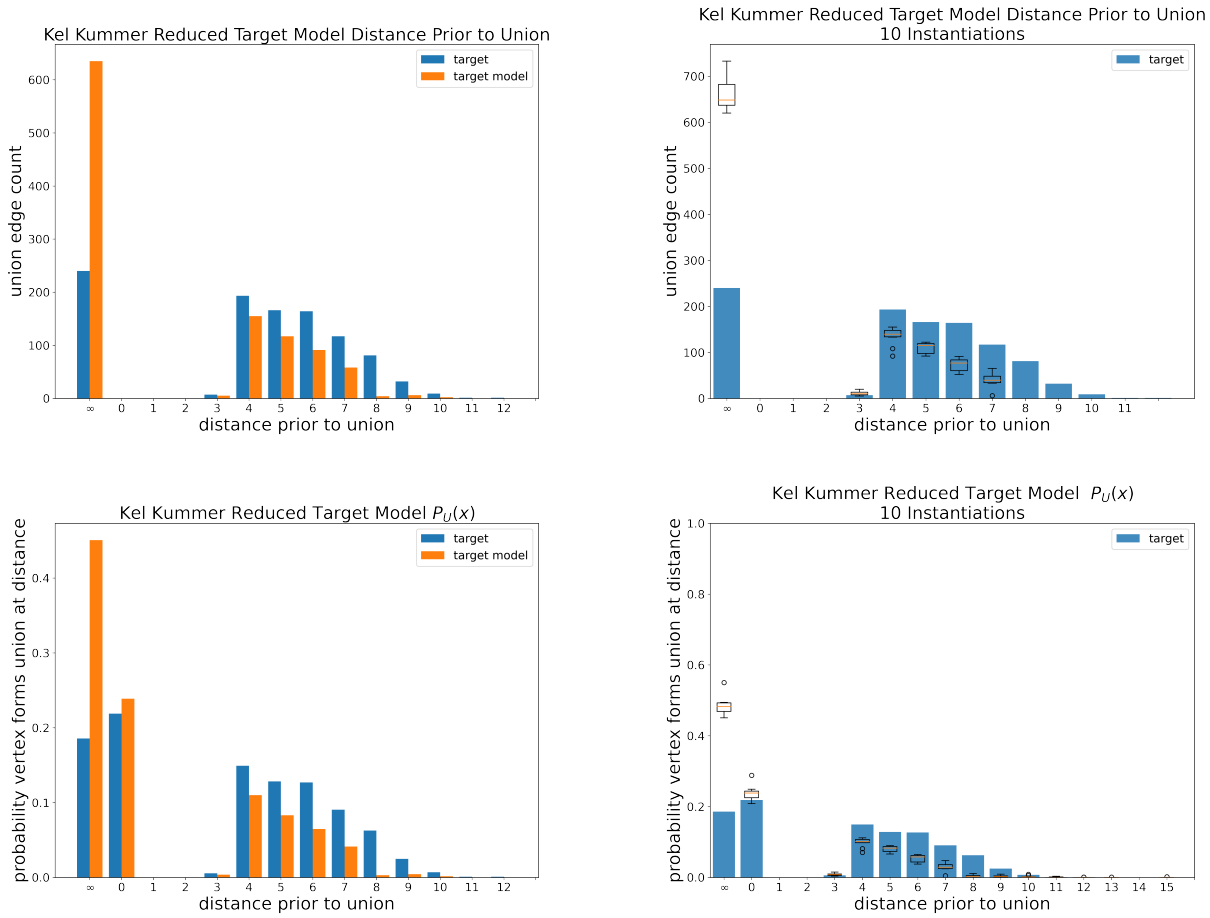

Figure 6.5: We run the Target Model as presented in Section 4.6, then omit the auxiliary ancestry which precedes the initial generation g_0. Omitting these paths to initial common ancestors changes the structure of the graph and results in the following comparisons for the Kel Kummer dataset. Top Row: The Target Model without auxiliary ancestry approximately captures the distribution of finite-distance unions, however by removing the common ancestry before g_0, the graph then misses the proportion of infinite-to-finite unions quite significantly. These distributions show the probability that a randomly selected union edge in the graph will be a certain distance from their partner. Bottom Row: The Kel Kummer dataset requires an initial population g_0 of 432 individuals. The Target Model introduces approximately 450,000 auxiliary vertices to impose a relative distance prior to union for each pairing in this initial generation. If we omit these auxiliary vertices, retaining only those individuals from our initial generation to our final generation $g_0, g_1, ..., g_L$, inclusive, then approximately 2600 vertices remain (our target size n_{target} for the Kel Kummer dataset is 2588 vertices). With approximately the correct number of vertices in the graph, the probability that a randomly-selected vertex in the modeled graph will be partner to a union at each specific distance approximately mirrors that observed in the target graph, again with the exception that the the Target Model without auxiliary ancestry overshoots the proportion of infinite-distance unions and undershoots the proportion of finite-distance unions. Contrast these results with those presented in Figure 6.1, where our calculations of distance prior to union include the auxiliary structure.

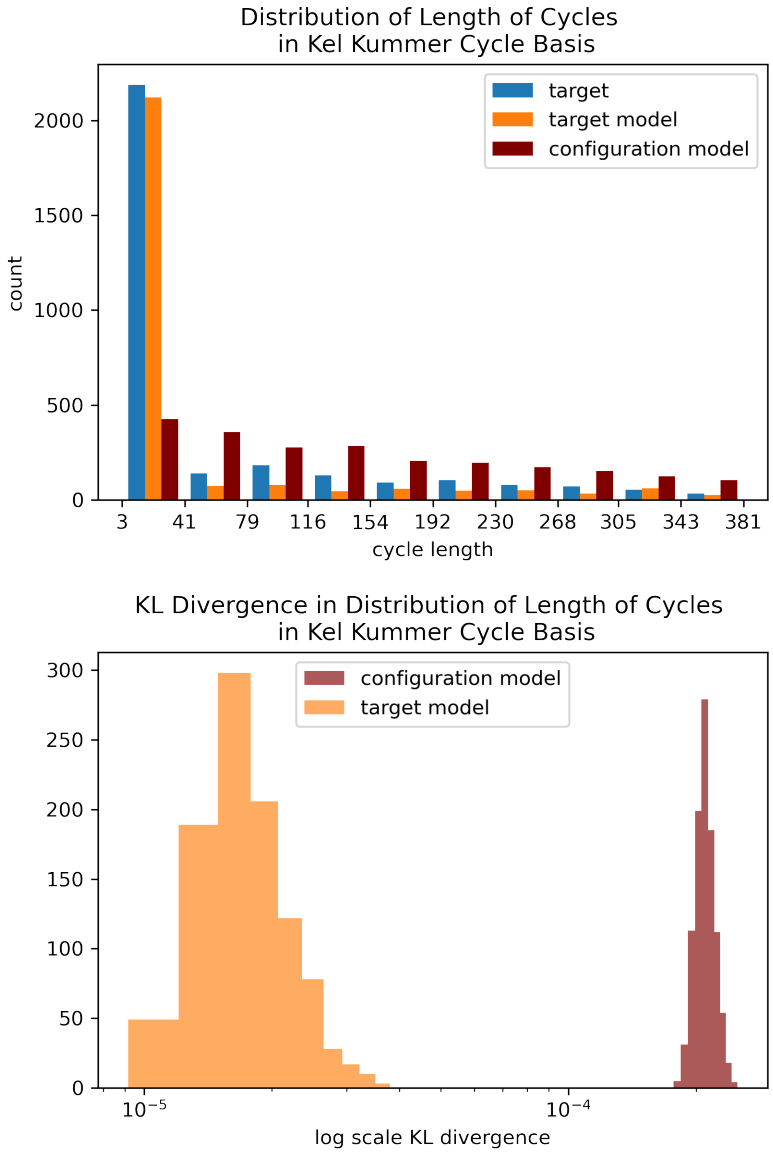

Figure 6.6: We run the Target Model as presented in Section 4.6, and then omit the auxiliary ancestry preceding the initial generation g_0 before calculating cycle bases. Top: For a single instantiating of the Target Model (omitting auxiliary ancestry) using the Kel Kummer dataset, we compare the lengths of the cycle basis of the undirected real-world network (blue) against that of an undirected Target Model (orange). We further compare the Target Model against a configuration model, created using the Target Model's degree distribution. Notice that the Target Model's cycle basis more nearly approximates the distribution of lengths of cycle basis elements in the real-world target network. Bottom: For 1000 instantiations of the Target Model (omitting auxiliary ancestry after completion) using the Kel Kummer dataset, we form a configuration model based on the Target Model's degree distribution. We then measure the lengths of elements in the cycle basis for the Target Model and for the configuration model, then we measure the KL divergence between the Target Model's distribution of lengths of cycle basis elements against that of the target network. We do the same for the configuration networks. These histograms show that the Target Model's distribution of cycle lengths more nearly approximates the distribution of cycle lengths in the target network. Contrast these results with those in Figure 6.3, where we retain auxiliary structure when calculating cycle bases.

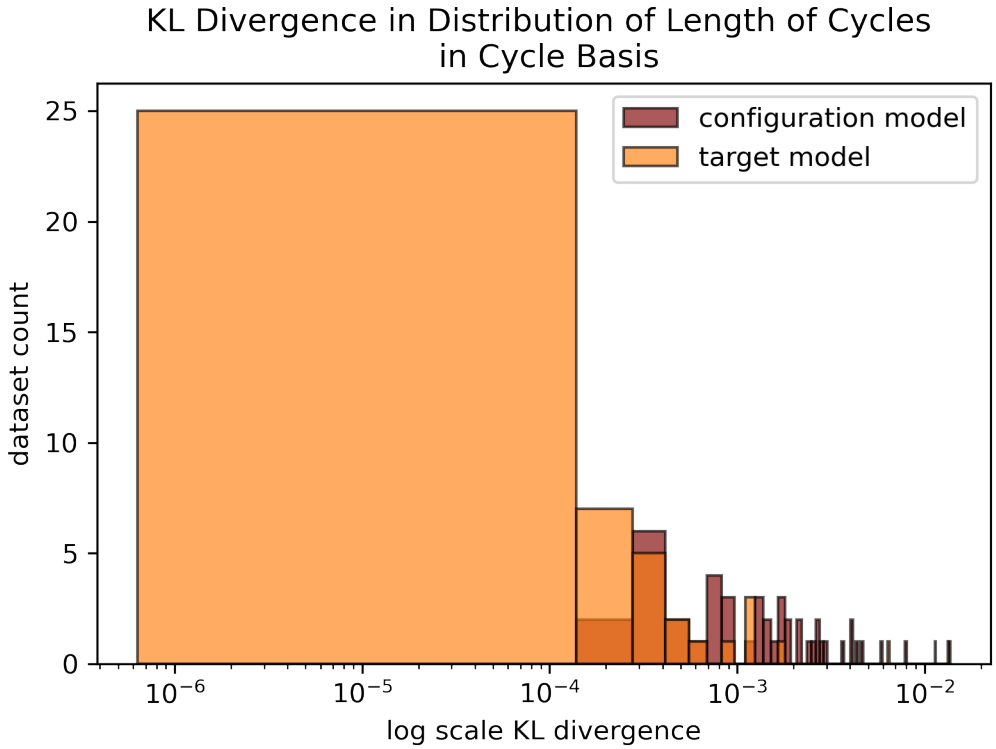

Figure 6.7: For each of the real-world target networks (see Appendix A), we instantiate 1000 Target Models, omitting the auxiliary ancestral lines preceding the initial generation g_0 after the Target Model executes as in Algorithm 4.6. For each of the Target Models, we instantiate a configuration model. For each of the 1000 Target Models and for each of the 1000 configuration models, we find a cycle basis, and measure the length of each element in the basis to form distributions of lengths of cycle basis elements. We then measure the KL divergence between each Target Model graph's distribution of lengths of cycle basis elements and that of the target real-world network. We take the median KL divergence score from the set of 1000. We treat the configuration models similarly, finding the KL divergence between the target network's distribution of lengths of cycle basis elements and that of each of the 1000 configuration models, recording the median performance. A smaller KL divergence indicates better agreement between distributions and we note that almost universally the Target Model produces networks whose cycle bases are more similar to the cycle bases of the corresponding real-world genealogical networks.

6.3 Variant Target Model—Results Omitting Auxiliary Ancestry Before Model Growth

The Variant Target Model (see Chapter 5) never connects pairings in the initial generation g_0 to any common ancestry. In the Variant Target Model all vertices in g_0 are an infinite distance away from one another so that only infinite-distance unions may form from g_0 pairings. The model then grows, adding children to form g_1, then forming unions from g_1 pairings and so on. This modeling choice creates a burn-in period. A certain number of generations must be added to the graph before finite-distance unions may form. This Variant Target Model is highly volatile. Many of our datasets require such a large starting population that the algorithm meets its stopping criteria after introducing only a single additional generation of children vertices. This is the case for the Kel Kummer dataset, which requires such a large number of initial vertices that the modeled graph has a very large initial generation, forms unions among this initial population, introduces children to these unions, and then the algorithm exits (see Figure 6.8).

For other datasets, such as the Torshan genealogical network from Mauritania in West Africa, the Variant Target Model produces graphs which have a larger number of generations. In this case, the Variant Target Model produces networks which approximate the size of the target dataset and which approximately capture the target distance to union distribution $P_U(x)$ (see Figure 6.10). The target number of children per house hold distribution is again constructed with independent identically distributed random samples and so is well approximated, given a sufficiently large modeled network.

Once again, networks produced by the Variant Target Model do better than configuration models of the same class in approximating the distribution of lengths of fundamental cycles in the target dataset (see Figure 6.11). That networks produced by the Variant Target Model tend to outperform their corresponding configuration models in terms of mimicking the target network's fundamental cycle lengths more closely (see Figure 6.12) provides an indication that the length of fundamental cycles alone is an insufficient metric to capture

the structure of a genealogical network. Consider that the Variant Target Model for the Kel Kummer dataset fails to capture the distance to union distribution $P_U(x)$ (see Figure 6.8) yet its distribution of fundamental cycle lengths still has a lower KL divergence from that of the real-world Kel Kummer dataset than configuration models of the same degree distribution (See Figure 6.9). It is insufficient that cycles of the correct lengths appear—in order for a modeled network to look like a genealogical network those cycles must occur in the correct locations, as captured by the distance to union distribution $P_U(x)$. While the Kel Kummer Variant Target Model produces shorter fundamental cycles such as appear in the real-world network, these cycles occur between two generations only. Contrast this with the cycles formed in a modeled network with more generations, such as for the Torshan dataset. These cycles are centered at unions and extend across many generations, which is much more like the real world dataset.

6.4 Conclusion

We have presented modeling techniques for genealogical networks. The Target Model (see Algorithm 4.6) forms networks which approximate a real-world genealogical network's distribution of distance prior to union $P_U(x)$ and its distribution of the number of children per household $P_C(x)$. These two metrics capture both the global and the local network structure, respectively, and largely account for the structure of a genealogical network. We have also discussed variations on this algorithm and have demonstrated that each modeling choice comes with both advantages and disadvantages. The Target Model with auxiliary ancestry (Algorithm 4.6) captures $P_U(x)$, given that we only examine the distribution in terms of union edges rather than in terms of the vertices in the graph. Additionally, this main algorithm results in networks which often dwarf their real-world counter parts. If we run the Target Model and then remove the auxiliary ancestral lines, the resulting networks are approximately the same size as the target real-world networks, but at the cost that we form a larger proportion of infinite-distance unions than desired. The Variant Target Model

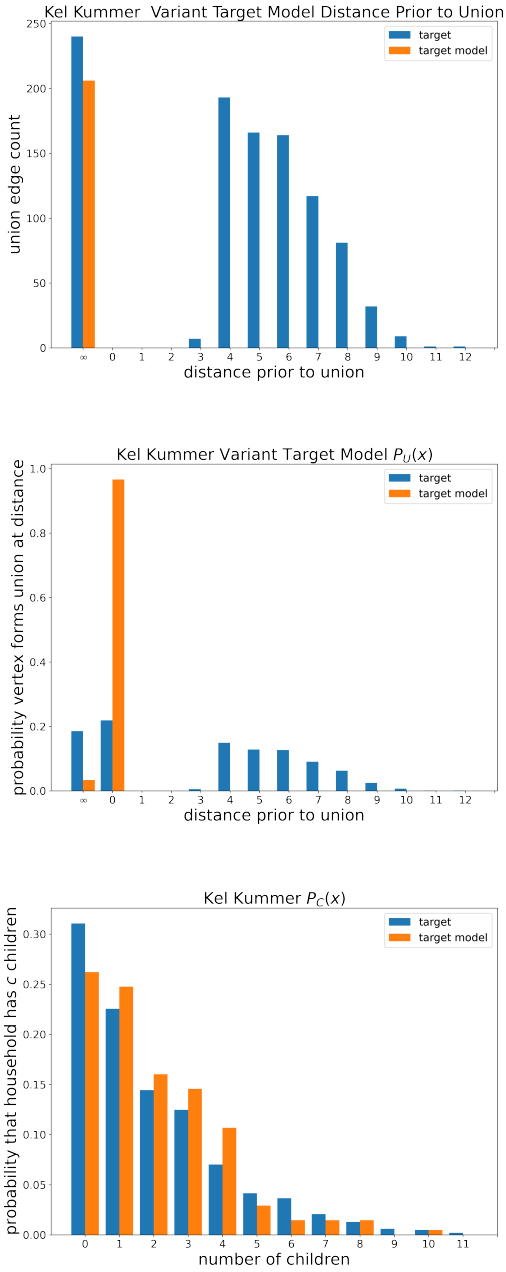

Figure 6.8: The Variant Target Model (see Chapter 5) for the Kel Kummer dataset requires a very large starting size. In fact, the model dies out with starting sizes below $n_0 = 2216$, some 85% of the real world Kel Kummer network's $n_{target} = 2588$ vertices. This is such a large initial generation that the model executes only a single iteration—forming infinite distance unions among pairings of vertices in g_0 and introduces children to these households—before the modeled network surpasses n_{target} in size. As a result, our model has only two generations, and all unions are at an infinite distance, while most vertices do not form unions.

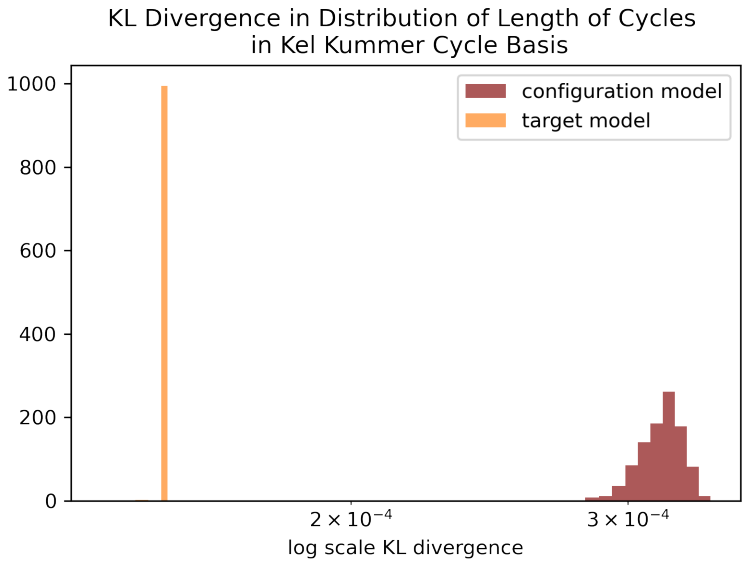

Figure 6.9: The Variant Target Model (see Chapter 5) for the Kel Kummer dataset requires a very large starting size. In fact, the model dies out with starting sizes below $n_0 = 2216$, some 85% of the real world Kel Kummer network's $n_{target} = 2588$ vertices. This is such a large initial generation that the model forms infinite distance unions among pairings of vertices in g_0 and introduces children to these households before the modeled network surpasses n_{target} in size. Even though our modeled networks generally have only two generations and all unions are at an infinite distance, the Variant Target Model for the Kel Kummer dataset still out performs its configuration model counterparts in terms of approximating the distribution of lengths of fundamental cycles in the real-world Kel Kummer dataset. Knowing that the Variant Target Modeled networks have fewer generations than the real genealogical dataset shows that fundamental cycle lengths alone does not capture the full structure of the real-world genealogical network. It is insufficient that cycles of the correct lengths appear—in order for a modeled network to look like a genealogical network those cycles must occur in the correct locations, as captured by the distance to union distribution $P_U(x)$.

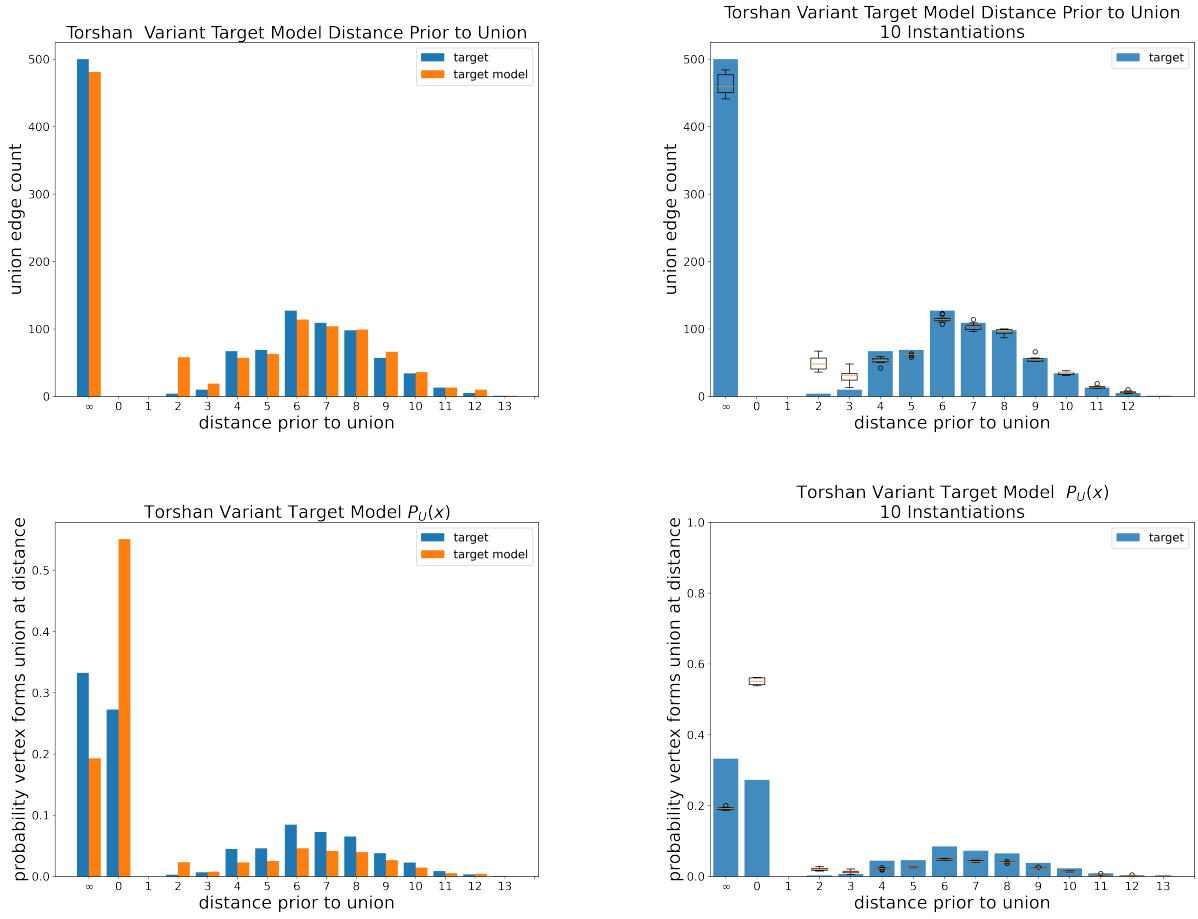

Figure 6.10: The Variant Target Model (see Chapter 5) for the Torshan dataset requires only a modest initial population $n_0 = 380$, some 12% of the real-world Torshan network's $n_{target} = 3008$ vertices. The Variant Target Model captures the Torshan dataset's distance to union distribution, both when conditioned on a vertex's being connected to union edges (Top Row) and when not conditioned on unioning (Bottom Row). The Variant Target Model for the Torshan dataset approximately mimics the size of the target network and the target distribution of distance prior to union $P_U(x)$.

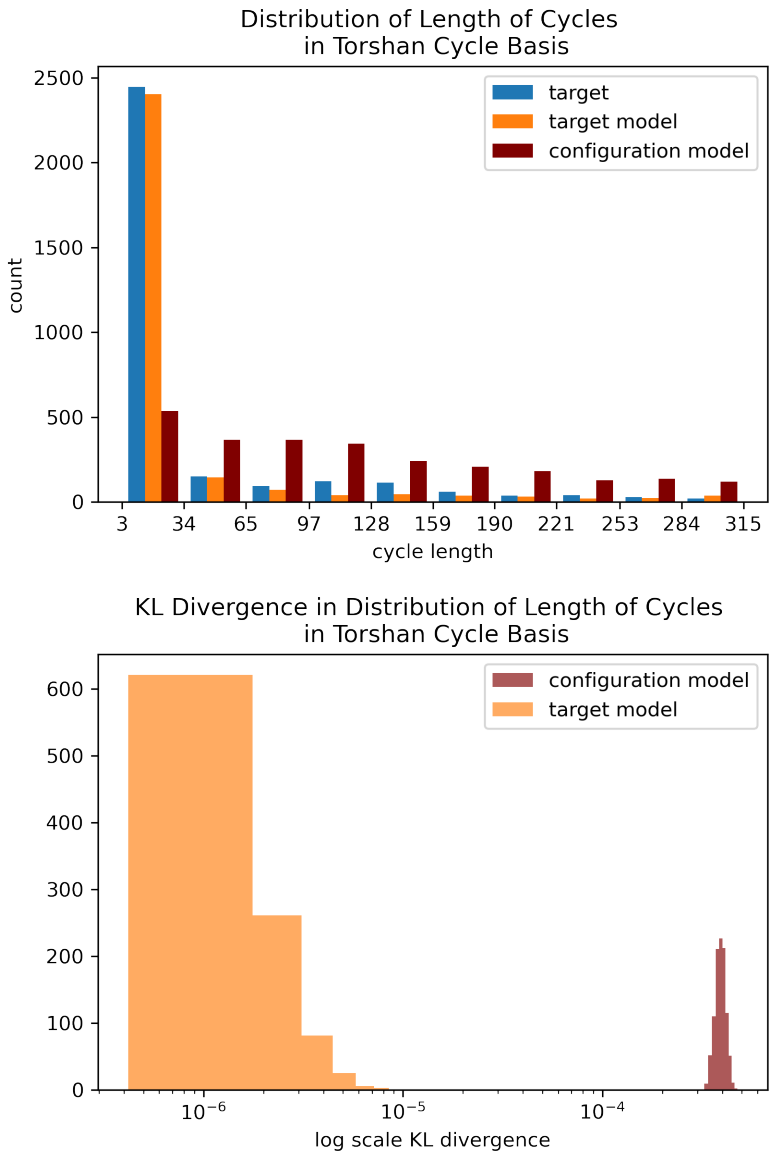

Figure 6.11: We run the Variant Target Model as presented in Chapter 5 for the Torshan dataset. Top: For a single instantiation of the Variant Target Model using the Torshan dataset, we compare the lengths of the cycle basis of the undirected real-world target network (blue) against that of an undirected Target Model (orange). We further compare the Variant Target Model against a configuration model, created using the Target Model's degree distribution. Notice that the Variant Target Model's cycle basis more nearly approximates the distribution of lengths of cycle basis elements in the real-world target network. Bottom: For 1000 instantiations of the Variant Target Model using the Torshan dataset, we form a configuration model based on the Target Model's degree distribution. We then measure the lengths of elements in the cycle basis for the Variant Target Model and for configuration model, then we measure the KL divergence between the Variant Target Model's distribution of lengths of cycle basis elements against that of the target network. We do the same for the configuration networks. These histograms show that the Variant Target Model's distribution of cycle lengths more nearly approximates the distribution of fundamental cycle lengths in the target network.

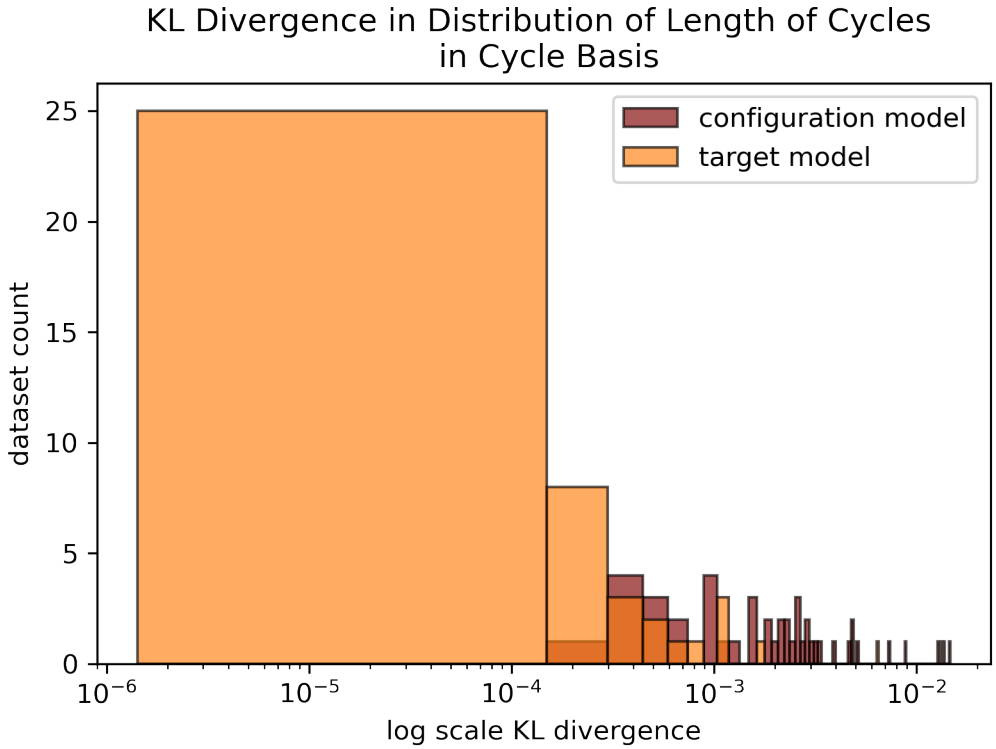

Figure 6.12: For each of the real-world target networks (see Appendix A), we instantiate 1000 Variant Target Models. For each of the Variant Target Models, we instantiate a configuration model using the degree distribution of each Variant-Target-Model produced graph. For each of the 1000 Variant Target Models and for each of the 1000 configuration models, we find a cycle basis, and measure the length of each element in the basis to form distributions of lengths of cycle basis elements. We then measure the KL divergence between each Variant Target Model graph's distribution of lengths of cycle basis elements and that of the target real-world network. We take the median KL divergence score from the set of 1000. We treat the configuration models similarly, finding the KL divergence between the target network's distribution of lengths of cycle basis elements and that of each of the 1000 configuration models, recording the median performance. A smaller KL divergence indicates better agreement between distributions and we note that almost universally the Variant Target Model produces networks whose cycle bases are more similar to the cycle bases of the corresponding real-world genealogical networks.

(Chapter 5) never introduces common ancestral lines to the initial population. This variant algorithm instead forms infinite-distance unions only until finite-distance unions are possible and often results in shallow networks with a small number of very large generations but in some instances produces modeled networks which look very like their real-world counterparts.

All three variations of the Target Model produce networks which do better than random (when compared against configuration models) at approximating the lengths of the fundamental cycles in the target real-world network. The three variations differ in their ability to capture the correct distance to union distribution $P_U(x)$. This shows that the lengths of fundamental cycles alone is insufficient to describe a genealogical network's structure and argues in favor of using distance to union and number of children per household to capture both global and local structure in genealogical networks.

Appendix A. Kinsources.net Genealogical Datasets

Table A.1: Genealogical Network Datasets.

Network Data			
Network Type & Name	Vertices	Edges	Citation
Genealogical Networks			
Datasets That Work with Our Model			
Genealogical Network 173	1140	2014	https://www.kinsources.net/kidarep/dataset-173-achuar-huasaga-chankuap.xhtml
Genealogical Network 150	795	1387	https://www.kinsources.net/kidarep/dataset-150-achuar-pastaza.xhtml
Genealogical Network 22	216	378	https://www.kinsources.net/kidarep/dataset-22-ainu-1880-as01.xhtml
Genealogical Network 3	659	1288	https://www.kinsources.net/kidarep/dataset-3-anuta-1972.xhtmlj

Network Name & Type	Vertices	Edges	Citation
Genealogical Network 92	636	1151	https://www.kinsources.net/kidarep/dataset-92-chenchu-1940-as02.xhtml
Genealogical Network 62	278	464	https://www.kinsources.net/kidarep/dataset-62-dogrib-1911-25-59-nd04.xhtml
Genealogical Network 65	645	1097	https://www.kinsources.net/kidarep/dataset-65-igluligmiut-1961-nu07.xhtml
Genealogical Network 164	128	114	https://www.kinsources.net/kidarep/dataset-164-kaingang.xhtml
Genealogical Network 34	502	786	https://www.kinsources.net/kidarep/dataset-34-netsilik-1922-nu09.xhtml
Genealogical Network 306	1463	1969	https://www.kinsources.net/kidarep/dataset-306-nobles-ile-de-france-1000-1440.xhtml
Genealogical Network 229	798	1416	https://www.kinsources.net/kidarep/dataset-229-nucoorilma-tingha.xhtml
Genealogical Network 251	619	1224	https://www.kinsources.net/kidarep/dataset-251-nunivak.xhtml
Genealogical Network 58	371	718	https://www.kinsources.net/kidarep/dataset-58-ojibwa-1930-nd07.xhtml
Genealogical Network 19	479	830	https://www.kinsources.net/kidarep/dataset-19-ojibwa-1949-nd08.xhtml
Genealogical Network 7	815	1582	https://www.kinsources.net/kidarep/dataset-7-pakaa-nova.xhtml
Genealogical Network 213	277	516	https://www.kinsources.net/kidarep/dataset-213-sarmi.xhtml
Genealogical Network 20	868	980	https://www.kinsources.net/kidarep/dataset-20-saudi-royal-genealogy.xhtml
Genealogical Network 18	294	441	https://www.kinsources.net/kidarep/dataset-18-tikopia-1930.xhtml
Genealogical Network 216	87	111	https://www.kinsources.net/kidarep/dataset-216-tiwi.xhtml

Network Name & Type	Vertices	Edges	Citation
Genealogical Network 242	125	202	https://www.kinsources.net/kidarep/dataset-242-tlingit.xhtml
Genealogical Network 11	169	275	https://www.kinsources.net/kidarep/dataset-11-top-of-the-mountain.xhtml
Genealogical Network 13	299	532	https://www.kinsources.net/kidarep/dataset-13-tory.xhtml
Genealogical Network 28	782	1366	https://www.kinsources.net/kidarep/dataset-28-trio-1960s.xhtml
Genealogical Network 41	48	86	https://www.kinsources.net/kidarep/dataset-41-vedda-1905-as04.xhtml
Genealogical Network 66	244	481	https://www.kinsources.net/kidarep/dataset-66-waimiri-atroari.xhtml
Genealogical Network 51	337	572	https://www.kinsources.net/kidarep/dataset-51-wilcania.xhtml
Genealogical Network 32	738	1212	https://www.kinsources.net/kidarep/dataset-32-yaraldi.xhtml
Genealogical Network 70	439	626	https://www.kinsources.net/kidarep/dataset-70-genesis.xhtml
Genealogical Network 258	1423	3211	https://www.kinsources.net/kidarep/dataset-258-todas.xhtml
Genealogical Network 115	4463	8416	https://www.kinsources.net/kidarep/dataset-115-charlevoix.xhtml
Genealogical Network 24	1269	2395	https://www.kinsources.net/kidarep/dataset-24-ayd-nl-yoruk-2005.xhtml
Genealogical Network 49	377	712	https://www.kinsources.net/kidarep/dataset-49-alyawarra-1971-au01.xhtml
Genealogical Network 223	1263	2021	https://www.kinsources.net/kidarep/dataset-223-samburu.xhtml
Genealogical Network 103	1695	3206	https://www.kinsources.net/kidarep/dataset-103-tikuna-arara.xhtml

Network Name & Type	Vertices	Edges	Citation
Genealogical Network 80	3008	6074	https://www.kinsources.net/kidarep/dataset-80-torshan.xhtml
Genealogical Network 158	240	395	https://www.kinsources.net/kidarep/dataset-158-tikar.xhtml
Genealogical Network 45	4178	7351	https://www.kinsources.net/kidarep/dataset-45-obidos.xhtml
Genealogical Network 78	147	242	https://www.kinsources.net/kidarep/dataset-78-puleliya-1954-simpler-version.xhtml
Genealogical Network 73	330	622	https://www.kinsources.net/kidarep/dataset-73-parakana.xhtml
Genealogical Network 87	105	245	https://www.kinsources.net/kidarep/dataset-87-arara.xhtml
Genealogical Network 89	116	220	https://www.kinsources.net/kidarep/dataset-89-nunamiut-1960-nu13.xhtml
Genealogical Network 287	4109	6517	https://www.kinsources.net/kidarep/dataset-287-duu-rea.xhtml
Genealogical Network 61	2588	5651	https://www.kinsources.net/kidarep/dataset-61-kelkummer.xhtml
Genealogical Network 128	3014	5454	https://www.kinsources.net/kidarep/dataset-128-ammonni.xhtml
Genealogical Network 249	5016	10719	https://www.kinsources.net/kidarep/dataset-249-baruya.xhtml
Genealogical Network 68	926	1951	https://www.kinsources.net/kidarep/dataset-68-surui.xhtml
Genealogical Network 35	2049	4159	https://www.kinsources.net/kidarep/dataset-35-chuukese-1947-1940.xhtml
Genealogical Network 30	2821	5079	https://www.kinsources.net/kidarep/dataset-30-manus-1929.xhtml
Genealogical Network 56	2477	4015	https://www.kinsources.net/kidarep/dataset-56-us-presidents.xhtml

Network Name & Type	Vertices	Edges	Citation
Genealogical Network 74	454	980	https://www.kinsources.net/kidarep/dataset-74-arawete.xhtml
Genealogical Network 54	3151	4289	https://www.kinsources.net/kidarep/dataset-54-feistritz-am-gael-1990.xhtml
Genealogical Network 44	585	1249	https://www.kinsources.net/kidarep/dataset-44-torres-strait.xhtml
Genealogical Network 93	9595	14988	https://www.kinsources.net/kidarep/dataset-93-sainte-catherine.xhtml
Genealogical Network 76	28586	51446	https://www.kinsources.net/kidarep/dataset-76-san-marino.xhtml
Genealogical Network 307	18645	32439	https://www.kinsources.net/kidarep/dataset-307-bwa-slam-biogsurvey.xhtml
Genealogical Network 194	8809	15643	https://www.kinsources.net/kidarep/dataset-194-kel-owey.xhtml
Datasets with 0 Finite Unions			
Genealogical Network 33	40	59	https://www.kinsources.net/kidarep/dataset-33-angmagsalik-1884-nu01.xhtml
Genealogical Network 10	80	132	https://www.kinsources.net/kidarep/dataset-10-apache-1932-nd01.xhtml
Genealogical Network 77	88	144	https://www.kinsources.net/kidarep/dataset-77-apache-1935-nd02.xhtml
Genealogical Network 204	399	592	https://www.kinsources.net/kidarep/dataset-204-dogon-konsogu-donyu.xhtml
Genealogical Network 39	118	192	https://www.kinsources.net/kidarep/dataset-39-eyak-1890.xhtml
Genealogical Network 31	17	24	https://www.kinsources.net/kidarep/dataset-31-family.xhtml
Genealogical Network 81	35	53	https://www.kinsources.net/kidarep/dataset-81-gundangborn-1948-au02.xhtml
Genealogical Network 37	178	274	https://www.kinsources.net/kidarep/dataset-37-igluligmiut-1921-nu05.xhtml

Network Name & Type	Vertices	Edges	Citation
Genealogical Network 226	116	176	https://www.kinsources.net/kidarep/dataset-226-jie.xhtml
Genealogical Network 5	105	172	https://www.kinsources.net/kidarep/dataset-5-konkama-1931-44-51-eu02.xhtml
Genealogical Network 14	168	221	https://www.kinsources.net/kidarep/dataset-14-labrador-inuit-1776-nu02.xhtml
Genealogical Network 17	218	353	https://www.kinsources.net/kidarep/dataset-17-lainiovouma-1952-eu03.xhtml
Genealogical Network 60	706	1177	https://www.kinsources.net/kidarep/dataset-60-mbuti-forest-1957-af02.xhtml
Genealogical Network 2	303	537	https://www.kinsources.net/kidarep/dataset-2-mbuti-village-1957-af03.xhtml
Genealogical Network 64	435	672	https://www.kinsources.net/kidarep/dataset-64-melombo.xhtml
Genealogical Network 12	90	119	https://www.kinsources.net/kidarep/dataset-12-miwuyt-1967-au03.xhtml
Genealogical Network 209	310	322	https://www.kinsources.net/kidarep/dataset-209-mowanjum-kalumburu.xhtml
Genealogical Network 21	19	30	https://www.kinsources.net/kidarep/dataset-21-ngatatjara-1966-au04.xhtml
Genealogical Network 42	304	472	https://www.kinsources.net/kidarep/dataset-42-nunamiut-tareumiut-1900-nu12.xhtml
Genealogical Network 75	98	161	https://www.kinsources.net/kidarep/dataset-75-nunamiut-1885-nu11.xhtml
Genealogical Network 79	139	201	https://www.kinsources.net/kidarep/dataset-79-paiute-1880-nd09.xhtml
Genealogical Network 223	1263	2021	https://www.kinsources.net/kidarep/dataset-223-samburu.xhtml
Genealogical Network 8	83	126	https://www.kinsources.net/kidarep/dataset-8-semang-1924-50-as03.xhtml

Network Name & Type	Vertices	Edges	Citation
Genealogical Network 4	95	157	https://www.kinsources.net/kidarep/dataset-4-shoshone-1860-nd10.xhtml
Genealogical Network 23	128	202	https://www.kinsources.net/kidarep/dataset-23-shoshone-1880-nd11.xhtml
Genealogical Network 69	77	134	https://www.kinsources.net/kidarep/dataset-69-slavey-1911-nd12.xhtml
Genealogical Network 171	219	371	https://www.kinsources.net/kidarep/dataset-171-suya.xhtml
Genealogical Network 38	20	28	https://www.kinsources.net/kidarep/dataset-38-wanindiljaugwa-1948-au06.xhtml
Datasets with Only 1 Finite Union			
Genealogical Network 52	378	609	https://www.kinsources.net/kidarep/dataset-52-apache-1936-nd03.xhtml
Genealogical Network 159	2975	5107	https://www.kinsources.net/kidarep/dataset-159-cocama-cocamilla.xhtml
Genealogical Network 84	48	76	https://www.kinsources.net/kidarep/dataset-84-hare-1956-nd05.xhtml
Genealogical Network 71	104	172	https://www.kinsources.net/kidarep/dataset-71-igluligmiut-1960-61-nu08.xhtml
Genealogical Network 240	410	746	https://www.kinsources.net/kidarep/dataset-240-kodiak.xhtml
Genealogical Network 90	1513	2217	https://www.kinsources.net/kidarep/dataset-90-omaha-1880.xhtml
Genealogical Network 15	112	182	https://www.kinsources.net/kidarep/dataset-15-oodnadatta.xhtml
Genealogical Network 91	64	109	https://www.kinsources.net/kidarep/dataset-91-takamiut-1927-64-nu03.xhtml
Datasets with Too Few Generations, Insuficient Structure			
Genealogical Network 36	272	445	https://www.kinsources.net/kidarep/dataset-36-copper-1922-nu10.xhtml

Network Name & Type	Vertices	Edges	Citation
Genealogical Network 46	29	48	https://www.kinsources.net/kidarep/dataset-46-hatfields-and-mccoys.xhtml
Genealogical Network 6	334	530	https://www.kinsources.net/kidarep/dataset-6-igluligmiut-1949-nu06.xhtml
Genealogical Network 9	289	477	https://www.kinsources.net/kidarep/dataset-9-konkama-1951-eu01.xhtml
Genealogical Network 48	367	671	https://www.kinsources.net/kidarep/dataset-48-wanindiljaugwa-1941-au05.xhtml
Genealogical Network 254	216	286	https://www.kinsources.net/kidarep/dataset-254-port-keats.xhtml
Genealogical Network 27	657	1166	https://www.kinsources.net/kidarep/dataset-27-nyungar.xhtml

Bibliography

[1] G. Rodriguez, "How genealogy became almost as popular as porn," *Time*, May 30, 2014. [Online]. Available: https://time.com/133811/how-genealogy-became-almost-as-popular-as-porn/

[2] FamilySearch. (2023) Familysearch.org facts. [Online]. Available: https://www.familysearch.org/en/newsroom/company-facts

[3] J. Greenwood, N. Guner, G. Kocharkov, and C. Santos, "Marry your like: Assortative mating and income inequality," *American Economic Review*, vol. 104, no. 5, pp. 348–353, 2014. [Online]. Available: https://www.aeaweb.org/articles?id=10.1257/aer.104.5.348

[4] I. Mathieson and A. Scally, "What is ancestry ?" *Plos Genetics*, vol. 16, no. 3, March 9, 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7082057/

[5] J. Companya Artes, J. Conesa Caralt, and E. Mayol, "Modeling genealogical domain: An open problem," in *International Conference on Knowledge Engineering and Ontology Development*, 2012.

[6] E. Malmi, A. Gionis, and A. Solin, "Computationally inferred genealogical networks uncover long-term trends in assortative mating," in *2018 World Wide Web Conference WWW*, Lyon, France, April 23–27, 2018, pp. 883–892.

[7] J. T. Chang, "Recent common ancestors of all present-day individuals," *Advances in Applied Probability*, vol. 31, no. 4, p. 1002–1026, 1999.

[8] V. Batagelj and A. Mrvar, *Pajek Program for Analysis and Visualization of Large Networks Reference Manual*, 2nd ed., 2011.

[9] "Data structure and file formats for kinship data," 2019. [Online]. Available: https://www.kinsources.net/editorial/dataset_formats.xhtml

[10] Kinsources. [Online]. Available: https://www.kinsources.net/

[11] M. E. Newman, *Networks*, 2nd ed. Oxford university press, 2018.

[12] A. Chaventre. (2015, 1983) Kelkummer dataset. [Online]. Available: https://www.kinsources.net/kidarep/dataset-61-kelkummer.xhtml

[13] ——, *Evolution Anthropo-Biologique D'Une Population Touaregue*, ser. Travaux et documents. Presses universitaires de France, 1983. [Online]. Available: https://books.google.com/books?id=c1zXdsCLlcQC

[14] T. E. of Encyclopaedia Britannica, "Tuareg," 2023. [Online]. Available: https://www.britannica.com/topic/Tuareg

[15] Z. M. Boyd, N. Callor, T. Gledhill, A. Jenkins, R. Snellman, B. Webb, and R. Wonnacott, "The persistent homology of genealogical networks," *Applied Network Science*, vol. 8, no. 1, p. 15, Feb 23, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s41109-023-00538-7

[16] W. Research, "FindFundamentalCycles," https://reference.wolfram.com/language/ref/FindFundamentalCycles.html, 2014.