Network characteristics associated with rapid HIV transmission among people who inject drugs in a rural county in the United States: a modelling study

Alyson L Singleton

Advisor / Matthew Harrison
Second Reader / Brandon Marshall
April 15th, 2019

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Sciences with Honors in the Division of Applied Mathematics at Brown University

Contents

1 Abstract

Background

People who inject drugs (PWID) are at high-risk for acquiring HIV in the rural United States. In 2015, Scott County, Indiana experienced the largest HIV outbreak among PWID in a rural setting in the United States. Structural characteristics of injection drug use networks are of great importance to understanding the increasing incidence of HIV in this sub-population.

Methods

We used an agent-based model to simulate HIV transmission in the combined sexual and injection drug use network in Scott County, Indiana. We calibrated the model based on observed values of Scott County and a rural, injection drug use population in Kentucky. We simulated the progression of HIV through Scott County with 1,000 iterations, allowing the stochasticity of the agent-based model to drive differences in the simulated network structures. We measured ten network metrics on each network to analyze how structural network characteristics affect the cumulative number of new HIV infections.

Results

The model predicted 211 cumulative incident HIV infections (95% SI: 17–306) by the end of the five-year simulation period. Our primary finding is the strong, first order effect of main connected component size on the number of cumulative incident HIV infections. We predict that with every added individual to the main connected component, the number of incident infections will increase by 0.4524. Our secondary finding also revealed that decreases in the maximum degree of HIV infected agents at $t = 0$, main component density, and transitivity in the main component might result in less extreme epidemic behavior.

Conclusion

Network structure characteristics unique to injection-drug use networks, such as a large main component, relatively high degree distributions, high density, and presence of high amounts of transitivity, put communities of PWID at a particularly high-risk for rapid HIV transmission. Prioritizing the development of network-based intervention strategies that address these characteristics, such as creating methods to identify individuals that bridge sub-populations of the network, is imperative in supporting the health of these communities.

2 Introduction

2.1 HIV and Injection Drug Use

The Centers for Disease Control and Prevention estimates that 1 in 10 new HIV infections in the United States are attributed to injection drug use practices or to sexual contact and injection drug use [1]. The sharing of needles and syringes or other injection equipment facilitates the spread of HIV and other viral blood-borne viruses such as viral hepatitis. Globally, 17.8% of people who inject drugs (PWID) are living with HIV. In North America, 9.0% of PWID are estimated to be living with HIV, which equates to around 230,500 people [2].

Sharing injecting equipment is an efficient method for transmitting HIV in comparison to other transmission methods, such as sexual transmission [3]. A number of factors are associated with rapid HIV transmission in PWID populations, including a lack of awareness that HIV is a threat in the local setting, high frequency of needle/syringe sharing, restrictions on access to sterile injecting equipment and/or lack of access to syringe service programs, recent changes in drug usage patterns, and large, interconnected risk networks [3]. High HIV incidence rates in PWID communities have been reported in numerous settings across the globe during the past decade [3].

2.2 Interventions to Mitigate HIV Transmission among PWID

There are manifold interventions and prevention efforts that are shown to effectively address factors that are associated with rapid HIV transmission among networks of PWID. These interventions include syringe services programs (SSPs), medication-assisted therapy (MAT), scale-up of antiretroviral therapy (ART), pre-exposure prophylaxis, and others [3]. In conjunction with these individually administered programs, network-based interventions have been developed to make use of the specific structural network characteristics of injection drug use (IDU) networks. Examples of network interventions include employing influential network members to provide available sterile equipment to their social and drug-using contacts, identifying and targeting interventions towards members connecting sub-populations, and changing social norms associated with risk behavior [3, 4]. Settings without these interventions are at high risk for a severe HIV outbreak, and unfortunately these interventions are often not implemented even after an outbreak is detected.

Network-based interventions offer hope that the same network structures which facilitate particularly efficient spread of HIV also create the potential to develop and deploy highly effective prevention and intervention methods. To implement these interventions, we need to study the structure of injection drug use networks and how that structure affects HIV transmission.

2.3 Networks and Disease Spread

A injection drug-using network represents our population of interest. At its core, our network takes the form of a simple graph. Graphs are used to investigate complex systems of interacting nodes. Each node, or agent, is represented by a vertex and each interaction between two vertices is represented by an edge connecting them [5, 6, 7]. Our network type of interest is a risk network. In a risk network, the nodes represent individuals of interest and edges represent connections bearing disease transmission risk. We focus our study around two types of connections: injection drug use with shared needles/syringes, and unprotected sex.

It is well known that risk network structure directly influences the dynamics of disease spread [6, 7, 11]. Risk network studies were first based around sexually transmitted diseases and sexual networks due to particularities of behaviors in how people choose to select and interact with their sexual partner(s). Heterogeneous contact patterns offer variability in disease incidence and prevalence rates [6]. Many mathematical models of infectious diseases, including compartmental models, offer valuable insight into disease spread without explicitly forming a network of any kind [8]. However, network-based simulations provide the ability to investigate the direct effects of network structure, especially as that structure pertains to specific settings and communities. Network-based models define an aspect of disease risk that goes farther than individual-level behavior—they reflect a person's place within their network and their sociometric proximity to others [4]. We use an agent-based model, TITAN, to investigate the variability in the structure of our networks of interest.

2.4 Injection Drug Use Networks

Although network models continue to primarily be used for sexual networks, the current surge of outbreaks in PWID populations in the United States calls for their use in investigating PWID contact network structure. In the wake of the current opioid use crisis, the relationship between network structure and disease dynamics highlights the relevance of research specific to injection drug use networks. Injection drug use networks are unique in their structure as compared to sexual networks or other types of risk networks. Depending on the question of interest, a risk network is can be built through injection risk acts alone or through a combination of both injection risk acts and sexual risk acts. There can be a substantial synergy and dependence between sexual and injection-related risk behavior, encouraging researchers to often include sexual behaviors as a separate, distinct process in injection drug use network studies [4]. Lastly, risk between two contacts in a network are often both sexual and injection in nature.

An instructive example for a general difference between injecting and sexual networks are their typical degree distributions (i.e. the typical distribution of one's number of risk contacts). PWID can have a more extreme number of high-risk contacts than most people in sexual networks [9, 10]. In contrast, studies have found the duration of relationships between risk contacts to be relatively similar between sexual and injecting partners, something one might not expect as injection relationships are often viewed as only interactions between acquaintances [11].

People in injection networks often fill specific roles within the network. In addition to the number of high-risk contacts a PWID might have, their location in the network influences their risk for certain outcomes and their impact on others' outcomes. The nature of injection drug use often results in certain typologies of network structure. For example, injection networks often contain one large connected component (a set of people who are either directly or indirectly connected to each other) and minimal other sets of connected communities [4, 11]. This connected component often has a "core" inhabited by "core members" that are deeply nested in the network [4, 12]. PWID networks also often have multiple sub-populations within a network that are connected by "bridges." These sub-populations, also called microstructures, can acts as "incubators" for HIV inside of larger networks, preserving the presence of HIV within a larger population [10]. The connectors between the sub-populations can be the difference between the continuation of an outbreak and the end of one. If the bridge is able to block the transmission of HIV into another sub-population (e.g., because they are already infected but virally suppressed and thus unable to transmit the infection to others), they are called a "firewall" [13, 14]. Fig. 1 shows a situation in which the firewall effect could take place. If Agent A enters treatment and achieves viral suppression, Agent A will block the infection from moving from Agent B into Section C of the network, blocking the spread of disease.

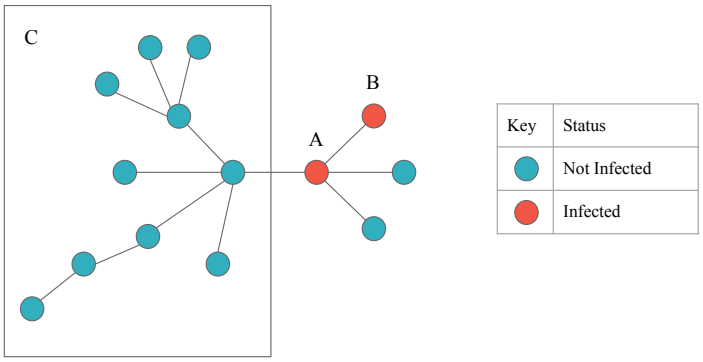


Figure 1: Firewall Dynamics Demonstration

We investigate the formal definitions of some of these network characteristics in this paper. Specifically, we investigate the influence of mathematically defined network metrics on the number of cumulative incident HIV infections in a model representing a PWID in Scott County, Indiana.

2.5 Scott County, Indiana

In 2015, Scott County, Indiana experienced the largest HIV outbreak among PWID in a rural setting in the United States. The community of PWID included an estimated 4,400 individuals out of the approximate 24,000 county members [9]. The sharing of injection drug use equipment for consumption of the prescription opioid oxymorphone use was determined to be the primary method of HIV transmission through this population [9, 15, 16]. Through self-reporting, 91.8% of HIV positive individuals and 86.6% of individuals in the high-risk contact network reported injection drug use [9].

The initial introduction of HIV into the high-risk PWID network is hypothesized to have occurred as early as 2011 by means of a sexual contact to a member of the PWID network [9]. The first diagnosis of HIV in this population was in November, 2014 [15]. By January, 2015, a public health alert identified the unusually high number of newly diagnosed HIV cases for Scott County at 11 (the previous nine years had produced a total of 5 new HIV diagnoses) [15,16]. In March, 2015, a public health emergency was declared, and in the following month both a local HIV clinic and syringe exchange program were created [15]. By September 2016, a total of 205 people had been diagnosed with HIV [9, 15].

Many PWID community members reported injecting between four and up to fifteen times daily. They also reported sharing injecting equipment in the form of syringes, cookers and crushing equipment [16]. Additionally, PWID reported preparing and using multiple injections per injection episode. The crush resistant technology in the extended-release formulation of oxymorphone drove PWID towards methods of preparation that left them with ¿1 mL of solution and syringes capable of holding only 1 mL. The need to inject all of the solution led PWID to complete multiple injections per injection episode–an act that increases the HIV risk [16]. This outbreak is a case study of a community in a historically low-risk HIV setting, where injection drug use facilitated rapid HIV transmission through the PWID population.

2.6 Question of Interest

Using the collected information about Scott County, Indiana and the county's injection drug use community, we built an agent-based model that represents the setting of Scott County. We use Scott County as a case study to investigate the network structure of IDU networks and their relationship to the severity of HIV outbreaks. We hypothesized that certain network characteristics would be associated with more severe HIV outbreaks, even within the same setting parameters and same population of PWID. We formed 1,000 different IDU networks representative and compared which network characteristics influence the number of cumulative incident HIV infections.

3 Methods

3.1 TITAN Model

This model of Scott County, Indiana was constructed through the work of William Goedel, Ph.D. Candidate at Brown University, for his paper, "Implementation of Syringe Services Programs to Prevent Rapid HIV Transmission in Rural Counties in the United States: A Modeling Study." The following is an extension of his work and an explanation of the modeling methods used to do the analysis of structural network characteristics in Scott County.

Setting

TITAN, Treatment of Infection and Transmission in Agent-Based Networks, is an agent-based model that was created by the Marshall Research Group of the Brown University School of Public Health. TITAN simulates the behaviors and interactions of autonomous agents. These micro-level interactions between agents drive the emergence of macro-level trends. We used TITAN to simulate HIV transmission within the 24,110 person population in the rural setting of Scott County, Indiana. The simulation lasts five years and progresses through month long time-steps. Agents are only able to leave the population either by death or by aging out at 65 years of age. Outside of this process, the agents remain in a static network for the entirety of the simulation. Our model was calibrated using observed Scott County data, which was supplemented from literature from comparable settings where necessary. All parameter values, key assumptions, and sources can be found in supplemental file of the first Scott County paper. This paper has been accepted by Clinical Infectious Diseases and will be cited here when it is released.

Agent Behavior and Network Formation

Injection Behavior and Injection Network Formation

The prevalence of injection drug use in Scott County is estimated at 1.7% [15]. Agents assigned as injection drug users are given a target number of injection partners per year from a negative binomial distribution with a mean of five partners [9]. IDU agents are also assigned a target number of injection acts per time-step (per month) from a Poisson distribution with a mean of 150 injection acts [17]. On average, 34% of these injection acts are estimated to include syringe sharing between contacts [17]. We assume that individuals unaware of their positive HIV status will share syringes at the same rate as un-infected individuals. We also assume that the event of diagnosis with HIV will not change an agent's number of injection partners or frequency of injection acts [18]. The probability that two PWID are connected through sexual contact in addition to injection use contact in the course of twelve months is 12.9% [9].

Sexual Behavior and Sexual Network Formation

All agents are assumed to be able to participate in sexual behavior in the model. Each agent is given a target number of sexual contacts per year, pulled from a negative binomial distribution with mean 1 [9]. Individuals are also assigned an ideal number of condomless vaginal intercourse acts per month, drawn from a Poisson distribution with mean thirteen [19]. Sexual acts that

included the use of condoms are not explicitly simulated because of the assumption that they do not facilitate the spread of HIV. We assume that individuals unaware of their positive HIV status engage in condomless vaginal intercourse at the same rate as un-infected individuals, but that diagnosis will reduce their number of condomless vaginal intercourse acts by 27% per month [18].

HIV Transmission, Treatment and Progression

The probabilities of transmission from all risk acts are estimated from a recent meta-analysis [20]. Seroconversion sparks a period of increased infectiousness in an individual for the two subsequent time-steps—representing acute stage infection. During this stage of infection, the probabilities of transmission for both injection acts and sexual acts are increased to five times their original value [21]. Individuals infected with HIV progress to AIDS with monthly probabilities that are dependent on their involvement and success with antiretroviral therapy [22].

We assume that 53.6% of all agents in our model have been tested for HIV at some point in their lives prior to the start of the simulation [17]. As the simulation progresses, each agent has a 1.7% monthly probability of being tested for HIV [23]. The model includes the public health alert at ten new HIV diagnoses that incited the contact tracing investigation. This investigation increases the probability of testing to 87.3% for one monthly time-step for individuals whose partner was diagnosed with HIV in the previous time-step [15].

Individuals initiate antiretroviral therapy in the time-step post-diagnosis with a probability of 60.8% [15]. For diagnosed individuals, the probability of achieving viral suppression is 42.8% [24]. Individuals that successfully achieve viral suppression reduce their chance of transmission by 94% [25]. Individuals that are enrolled in treatment but not yet virally suppressed still achieve a reduction of transmission by 17% [25].

Model Scenario

The following scenario was simulated with 1,000 iterations. Each simulation models HIV transmission in the population of Scott County for five years from 2011 to 2015. Each iteration includes the formation of the model population, the formation of the network, the introduction of HIV into the injection drug use network through a randomly selected injection drug user, and the five-year progression of HIV throughout the population. Descriptions of these four processes follow.

Population Formation

The model is initialized with a virtual population of 24,110 individuals. These agents represent the portion of the Scott County population aged 18 to 64. The base population is seeded stochastically to represent the desired age, gender, and gender-specific prevalence rates of HIV-infection in the population. Behavioral and clinical variables are also assigned stochastically to each individual. The population size, age assignments, and appropriate prevalence rates are based on the 2006-2010 cycle of the American Community Survey by the U.S. Census Bureau [34]. The values of base level prevalence are derived from the number of people who were re-

ported to be living with HIV at the end of 2010. At the end of the decade, 22 people were living with HIV (17 males and 5 females). This corresponds to an overall prevalence of 0.09% (0.14% among males and 0.04% among females) [35]. Age-specific mortality rates were derived from aggregating data from 2006-2010, based on statistics for Scott County, Indiana. These statistics are provided in the WONDER database by the National Center for Health Statistics [36].

TITAN model makes use of three pseudorandom number generators (PRNGs). A PRNG is an algorithm that generates a sequence of numbers that approximates the properties of a random number sequence. The reason we use a PRNG rather than an random number generator (RNG) is because a PRNG's number sequence is completely determined by its initial value, called a seed. In other words, if a PRNG is reinitialized with the same seed, it will produce the same sequence of numbers. We use the first PRNG during population formation, during which this sequence of numbers to evaluate the "dice rolls" (i.e. the aforementioned stochastic processes).

Different demands of the seed value allow for purposeful introduction or limitation of noise in our scenarios. For this analysis, we fix the seed when we initialize the population during each of the 1,000 iterations. Thus, we are able to demand the same exact base population be created for each of our 1,000 simulations. However, we are also able to randomize the seed value, creating new base populations for each simulation. Note that all of these populations adhere to our input parameters, but vary slightly due to the aforementioned stochastic processes (i.e. there will be the same proportions of agents who are females and males, different ages, HIV infected and non-infected, but the combinations of these characteristics within individuals might differ). Lastly, we are able to demand that the seed progresses through a sequence of integers (i.e. 1, 2, 3, ...) so that the populations vary each run but remain reproducible for follow up analysis.

Network Formation

During initialization, a sexual and injection drug use network is formed using the simulated population. As agents are built, they are assigned a target number of injection and/or sexual partners. They are also assigned a target number of injection and/or sexual acts per time-step (per month). The risk network is formed through an iterative process. The process loops through the simulation population of n individuals. With each iteration, routines match those in need of a partner to others also in need of a partner. Eligible partners are added to the partner list associated with each individual based on their target number of partners. After looping through the entire set of network members, the pool is iterated and partners are paired using the pairing algorithm. We repeatedly loop through the simulated population in this manner until all individuals' target numbers of partners have been met. We define this as the equilibrium state of the network (i.e. even if we continue to run these routines, no changes will occur as no agent is still seeking any type of partnership). After this state is reached, ties are assumed to be static for the duration of the simulation. Estimations for the duration of sexual and/or injection partnerships in Scott County are unavailable. Instead, we employ the estimations measured in a sample of adults in Perry County, Kentucky, a rural setting similar to Scott County in population size, racial composition, education levels, and employment rates [37]. In this study, the average duration of partnership was 10.2 years [37]. Because the simulation duration is only

five years long, we assume the network remains static.

TITAN model uses its second PRNG for the network formation process. This PRNG is exactly similar in function to the PRNG used during population formation. Importantly, we do not hold this seed constant for each of the 1,000 iterations. We randomize this seed value so that a new risk network forms for each of the 1,000 times we run the scenario. Again, each of these networks respect the degree distribution of Scott County through the partnering process detailed above. However, the structure of the network is still variable due to the stochastic framework of the partnering. Even with the same base population, agent A might partner with agent B in one simulation, but in the next might partner with agent C instead. Here, we introduce noise into the model in the form of varying network structures. Each of the 1,000 generated networks represent a possible risk network of Scott County, but their differences allow us to capture the variability observed in injection-drug networks.

Infection Introduction

After the formation of the network, we randomly choose a single agent to spontaneously seroconvert. This introduces HIV into the network. The only specification for the recipient of the initial infection is that it must be a PWID. As some PWID reside outside of the main component, about 40 of our 1000 runs result in an outbreak that did not breach the main component. See Section 3.2, Subsection: Component Size, for a definition of main component. Recall that the network is static, preventing the infection from ever entering into the main component if the infection is not already present in the component at $t = 0$. Also recall that there is a baseline prevalence of HIV in the component before the infection is dropped into the IDU network. This results in some simulations where there is already an HIV-infected individual in the main component prior to the spontaneous seroconversion. Therefore, we occasionally see more than one HIV-infected agent in the main component. This is addressed further in Section 3.2, Subsection: Degree Centrality.

There is no PRNG that controls the variation of the infection introduction. However, adding a fourth PRNG could control whether the location of the spontaneous seroconversion varies or remains the same during runs with the same base population and same network structure. This offers an interesting opportunity for future work. See Section 5.3, Subsection: Variability and Noise, for further discussion.

Five-Year Progression

The final step is to let the simulation proceed through the five-year time period. At each time-step, a series of stochastic processes facilitates the relevant changes and developments in the virtual setting. These changes include the emergence of HIV through risk acts between partnerships, the population's engagement with HIV prevention and treatment services (e.g., treatment as prevention, condom use, pre-exposure prophylaxis, syringe services programs), the progression of HIV within those that acquire the infection, and behavior changes due to change in personal or partner status (e.g. sexual risk acts diminishing due to receiving a positive HIV

test). At each time-step, the simulation determines which changes to make and applies them. The model progresses in this fashion for five years (sixty month-long time-steps). In termination, the simulation outputs our final outcomes of interest.

The third and final PRNG is used in this step. We allow this seed to vary randomly as in the network formation process. We only run the five-year progression once for each network. Therefore, this seed value has little impact on our analysis, though it would be useful if we wanted to run each network multiple times. Again, see Section 5.3, Subsection: Variability and Noise for further discussion.

3.2 Network Analysis

Networks as Graphs

As mentioned in the introduction, our risk network, at its most base level, takes the form of a simple graph. The entire grouping of nodes and edges is defined as a simple graph. Indeed, we formally define a *simple graph* as an ordered pair:

$$G = (V, E) \tag{1}$$

where V is a non-empty of vertices and E is a set of edges. Let us define the sets

$$V = \{v_1, v_2, \ldots, v_n\}$$

$$E = \{e_{i,j} \mid i, j = 1, \ldots, n\}$$

where $e_{i,j}$ is the edge connecting v_i and v_j. We specify that $i \neq j$ and that $e_{i,j} = e_{j,i}$. This means that we are operating in an undirected network and that vertices cannot be connected to themselves–hence our specification that we are working with a *simple* graph. Our network type of interest is a risk network. In a risk network, the nodes represent individuals of interest and edges represent connections bearing disease transmission risk. We focus our study around two types of connections: injection drug use with shared needles/syringes and unprotected sex, both of which propagate risk of infection.

Let us also take the opportunity to define the adjacency matrix of our graph, G. We say that two vertices, v_i and v_j, are adjacent if they are connected by an edge, $e_{i,j}$. Define the $n \times n$ *adjacency matrix* of graph, G as

$$A = (a_{i,j}). \tag{2}$$

A notates all of the relationships between agents, or lack there of, as follows: if there is an edge from v_i to v_j, then $a_{i,j} = 1$ and we write 1 as the entry of row i column j of matrix A. If there is no such relationship, then $a_{i,j} = 0$ and we write 0.

Network Metrics

We use this section to define our network statistics of interest. We describe the statistic in words, define its mathematical notation, explain its epidemiologic importance, and specify how

we make use of the statistic in our network analysis.

Degree Centrality

The degree centrality of a vertex is the number of vertices that vertex is connected to by an edge (i.e. the number of neighbors that a specific vertex has). The *degree centrality*, d_i of vertex v_i is notated as:

$$d_i = \sum_j a_{i,j} \tag{3}$$

Recall that $a_{i,j}$ is the value in row i column k of A, the corresponding adjacency matrix.

If a vertex has higher degree, it is more influential than others in the network. A large number of connections is directly aligned with a large risk of being infected and of infecting others in a risk network [5]. Importantly, degree centrality only accounts for the effect of nearest-neighbors to a vertex. It does not tells us any further information about vertices of farther distance away from our vertex of interest. We measure two versions of degree centrality due to the aforementioned variability in number of HIV-infected nodes at $t = 0$ (described in Section 3.1, Subsection: Infection Introduction). First, we report the average degree centrality of all HIV-infected vertices in the main component at $t = 0$ (the grouping of vertices that play a role in starting the outbreak). Second, we report the degree centrality of the vertex that has the highest degree centrality of all HIV-infected vertices in the main component at $t = 0$ (the vertex we infer to be the most influential). Recall, if the degree is reported as zero, that means that there are no HIV infections in the main component. Again, see "Component Size" for definition of main component.

Betweenness Centrality

To define betweenness centrality we must first define two other terms. First, we define a *walk* of length l as any sequence of vertices $v_1, v_2, \ldots, v_l, v_{l+1}$ such that $\forall i = 1, 2, \ldots, l$ there is an edge $e_{i,j}$. Second, we define a *path* of length l as a walk of length l in which all the vertices (and all of the edges) are distinct. Now, we may define betweenness centrality of a vertex, v_k, as the fraction of shortest paths between some pair v_i and v_j that pass through v_k [26]. Let $\rho(i, j)$ be the number of shortest paths from v_i to v_j. Let $\rho(i, k, j)$ be the number of shortest paths from v_i to v_j that pass through v_k on their way. Then *betweenness centrality*, $BC(k)$ is notated as [1]:

$$BC(k) = \sum_i \sum_j \frac{\rho(i, j, k)}{\rho(i, j)}, \quad i \neq j \neq k \tag{4}$$

Formally, a sub-population is defined as a subgraph. A *subgraph* is defined as

$$S = (V', E') \tag{5}$$

iff $V' \subseteq V$ and $E' \subseteq E$ [5]. Betweenness centrality is used to assess the dependency of a disease on a certain vertex to reach all ends of the graph. If a vertex has a high level of betweenness centrality, it is integral to the passing of disease across a network. Nodes with high betweenness centrality distributions are often the connection between two sub-populations. A logical caveat to this metric is that not all communication moves along the shortest paths in a network. There

are other definitions of betweenness that address this discrepancy. However, in our epidemiologic setting, the shortest path is especially relevant as every one of our graph's edges represent risk. The risk is often weighted differently depending on the class of connection, but the assumption that the shorter the distance between vertices, the more likely the disease will reach one from the other. We measure the average betweenness centrality of the main connected component.

Random Walk Betweenness Centrality

Random walk betweenness centrality is the number of times a random walk from v_i to v_j passes through v_k averaged over all i and j [26]. To define random walk betweenness centrality first we must define the Laplacian matrix. Recall that we call the adjacency matrix for our simple graph, A. Next, we define the *degree matrix* as D, where D is the diagonal matrix with elements $D_{i,i} = d_i$. We define the *Laplacian matrix* , L, as $L = D - A$. Recall that we have defined G as a simple graph, forcing A to contain only 0s and 1s and to have diagonal elements of 0. To calculate the *random walk betweenness centrality* [5, 26],

1. Construct the Laplacian matrix, L for our graph, G.
2. Remove a row k and its corresponding column k, to obtain an $(n-1) \times (n-1)$ matrix L_k.
3. Define matrix T as the matrix L_k^{-1} with both a row and column of zeroes added back into the position from which the row and column were first removed. Call the elements $T_{i,j}$.
4. Define
$$I_r^{pq} = \frac{1}{2} \sum_j A_{rj} |T_{rp} - T_{rq} - T_{jp} + T_{jq}|, r \neq p, q \tag{6}$$
where $I_p^{pq} = I_q^{pq} = 1$.
5. Finish by using the values from Step 4 to evaluate:
$$BC_{RW}(r) = \sum_{i<j} I_r^{pq} \tag{7}$$

As we mentioned earlier, there are many definitions of betweenness. This is an additional measure of betweenness that we can compare to the betweenness metric defined as $BC(k)$ in (4). Random walk betweenness centrality offers valuable comparison to BC. With BC, we are working under the intuition that the disease "knows" the the shortest, most efficient way to reach its target . With BC_{RW} the disease moves at random through the population of interest. If we walk randomly from v_i to v_j many times and we pass through v_k often, that indicates that v_k plays a significant role in the transmission of disease from v_i to v_j. By averaging over the entire graph, G, we obtain an overall centrality measure for v_k. Shortest-path and random-walk are opposite in sentiment, and therefore are valuable to measure together [27]. We measure the average random walk betweenness centrality of the main connected component.

Component Size

First, let us define a *connected component* of a graph as a subgraph, S, of a simple graph G in which every vertex is connected to every other vertex in S by a path [5]. Recall the definition of a subgraph from (5). We also specify that none of the vertices, V', in subgraph S are connected to any elements of $(V')^c$ in the greater network. We often refer to a connected component solely as a *component*. Note that a vertex, v_i, with no corresponding edges, $\{e_{i,j}\} = \varnothing$, is, in and of itself, a connected component. We define *component size* as the number of vertices in set V'. Formally,

$$CS = |V'| \tag{8}$$

We often refer to a connected component solely as a component. Components reflect sub-populations of the network and represent that often these sub-populations are disconnected from one another. The size of a component effectively measures the maximum reach an infection in that component can obtain. Often sexual transmission networks are sparse and have multiple components [7]. However, the observed injecting network in Scott County is almost exclusively one large component [9]. We measure the component size of the main component. We define the main component as the largest component in the network. This component contains the majority of HIV transmission in the simulation and is meant to be representative of the main observed component of injection drug users in Scott County.

Component Density

We begin by defining density of a graph in its entirety. The density of a graph is the number of edges in the graph divided by the number of total possible edges the graph might have. Formally, graph *density* is defined as [3, 10]:

$$D = \frac{2|E|}{|V|(|V| - 1)} \tag{9}$$

Component density has the same sentiment, except that it is restricted to the number of edges and possible edges in the component. Therefore, *component density* is written as:

$$D = \frac{2|E'|}{|V'|(|V'| - 1)} \tag{10}$$

Component density is the density of the main component of interest. Network, and therefore population, density largely determines the rate of infection [28]. We focus solely on component density as opposed to graph density because we specify that our network is static. Therefore, there is no chance of infection moving outside of this component, so we are only concerned with the density of the component where the infection is introduced. Importantly, density is not a perfect metric because it can be measured through an average of the vertices' densities, allowing for networks of varying degree distributions to reflect the same density value. However, it does provide a valuable, basic notion of the "knittedness" of the component of interest [29]. We measure the component density of the main component.

Geodesic Distance

The geodesic distance is the number of edges on the shortest path between two vertices. This notion of geodesic distance is also often referred to as the shortest path distance. Note that there may be more than one shortest path between two vertices. If the two vertices are in different connected components we say the distance between them is infinite. Formally, the average *geodesic distance* is [5]:

$$GD = \sum_{v_i, v_j \in V} \frac{d(v_i, v_j)}{(|V|)(|V|-1)} \tag{11}$$

where $d(v_i, v_j)$ is the shortest path from v_i to v_j in G [5]. Measuring geodesic distance partners well with measuring component density. While density reflects the sheer number of edges in the component of interest, geodesic distance speaks about the spread of the vertices [5]. We measure the average geodesic distance in the main component.

Centralization

The centralization of a network discloses how intensely the network is centralized around one or a few actors [10]. The maximum centralization value is 1 and occurs when the all vertices are connected to only one, central vertex. This situation is referred to as a star graph. The minimum centralization value is 0 and occurs when all vertices have the same degree value. This situation is referred to as a circle graph [30].

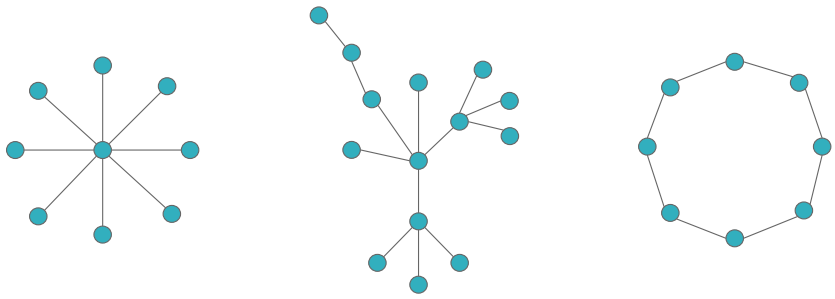


Figure 2: Star Graph: $C_d = 1$, Decentralized Graph: $0 < C_d < 1$, Circle Graph: $C_d = 1$

We define the *centralization* of a graph as [10, 30]:

$$C_d = \frac{\sum_{i=1}^{n}[d(v*) - d(v_i)]}{max \sum_{i=1}^{n}[d(v*) - d(v_i)]} \tag{12}$$

where d(v*) is the largest observed degree. This equation is also equivalent to the following definition. The following is what we use to calculate the measure in the python network analysis code [13]:

$$C_d = \frac{\sum_{i=1}^{n}[d(v*) - d(v_i)]}{(n-1)(n-2)} \tag{13}$$

If a graph, G, is an intermediate to a star graph and a circle graph, as our simulated Scott County networks are, then their centralization will be a value between 0 and 1. A high centralization measure indicates dependence on one or a few vertices as bridges between difference sub-populations. This is referenced as a hierarchical network and is anticipated to reduce HIV transmission (a phenomenon connected to the "firewall effect"). On the other hand, a low centralization measure implies that edges are evenly distributed across a network, creating an opportunity for disease to spread through multiple avenues.

K-Cores

First we must define a maximal subgraph. Recall the definition of a subgraph from (5). A graph, or subgraph, is *maximal* for a specific property if the graph has that property but no other subgraph of larger size has the same property [31]. In other words, it is the largest of the set of subgraphs with said property. With that definition in mind, a k-core is a maximal subgraph of G that contains vertices with degree at least k. Formally, we say that a subgraph $S = (V', E')$ is a *k-core* if

$$d(v_i) \geq k \quad \forall v_i \in V' \tag{14}$$

Note that the 1-cores of the graph are the non-trivial connected components [29]. Also note that there can be at most one 2-core in a connected graph [29]. We focus on 2-cores specifically because the 2-core of a risk network is considered a high-risk sociometric risk location [11]. Those individuals located in a 2-core are more likely to participate in high-risk behavior and are, predictably so, more at risk to acquire HIV [11]. We calculate the proportion of individuals located in 2-cores in the connected component.

Transitivity

The transitivity of a network is the proportion of all triads that exhibit closure in the network. A *triad* is the structure with two edges and a shared vertex. A *triad* that exhibits *closure* is a complete triangle. Formally, a triad is three vertices that form a complete graph (i.e. each vertex is connected to both other vertices with an edges). Therefore, the definition of *transitivity* follows [7]:

$$T = \frac{trace(A^3)}{||A^2|| - trace(A^2)} \tag{15}$$

where the *trace(A)* is the sum of the elements on the main diagonal of matrix A and $||A||$ is the matrix norm of A. The amount of transitivity in the network gives us an idea of the clustering of the network. This proportion measures the level of social cohesion happening in the network [7]. Vertices with high transitivity are more at risk for acquiring the disease because they are located in the more dense areas of the network and can be reached through multiple avenues. Diseases spread more readily through highly transitive networks [6]. We measure the the proportion of all triads in the entire network that exhibit closure.

Network Code

We made use of the NetworkX Python package to measure these structural characteristics. We wrote a python program to complete this task. First, the script alters the formatting of the edgelist from the TXT output by TITAN to a python dictionary. Then, the program analyzes the newly created edgelist and outputs a tab-delimited text file where each row represents a different iteration and each column represents the different network metric. This program therefore includes ten main functions, constructed through a combination of functions both from the networkX package and functions we edited and wrote on our own. You can find this python program in Appendix Section 7.4.

3.3 Statistical Analysis

Correlation Coefficient Analysis

We begin initial analysis of our results by constructing a primary set of graphs, each plotting the network metric values against the number of cumulative incident infections. These figures are included in Appendix Section 7.2. We investigate basic linear associations between each exposure and outcome through correlation coefficients. Correlation coefficients report the covariances between exposure vector and outcome value. They are measures of associations that can range from -1 to 1. Negative values represent a negative association and positive values represent a positive one. In a correlation test, the null hypothesis is represented by a correlation of 0, meaning there is no linear relationship. Specifically, for every unit increase in the network metric, we predict a corresponding increase or decrease of the coefficient value in our outcome. The formal definition of the sample Pearson correlation coefficient follows. Given paired data $(x_1, y_1), \ldots, (x_n, y_n)$ consisting of n pairs, where the x's are the metrics and the y's are the outcomes, we define r_{xy} as the following:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Note that n is the sample size ($n = 1000$) and $\bar{x}$ and $\bar{y}$ are the sample means.

First, we calculate the correlation coefficients between the values of each network metric and the corresponding number of incident infections. Next, upon reviewing the set of ten graphs, we notice a particularly strong relationship in the graph of main component size against the number of cumulative incident infections. There is a clear positive association of component size on the number of incident infections in the population. This specific graph is included below in Fig. 3 with a linear regression line. Component size appears to be most impactful exposure in comparison to the visualizations of the other nine exposure statistics. We consider that component size is most likely operating as a first order effect and is likely playing an underlying role in the other plots and coefficients.

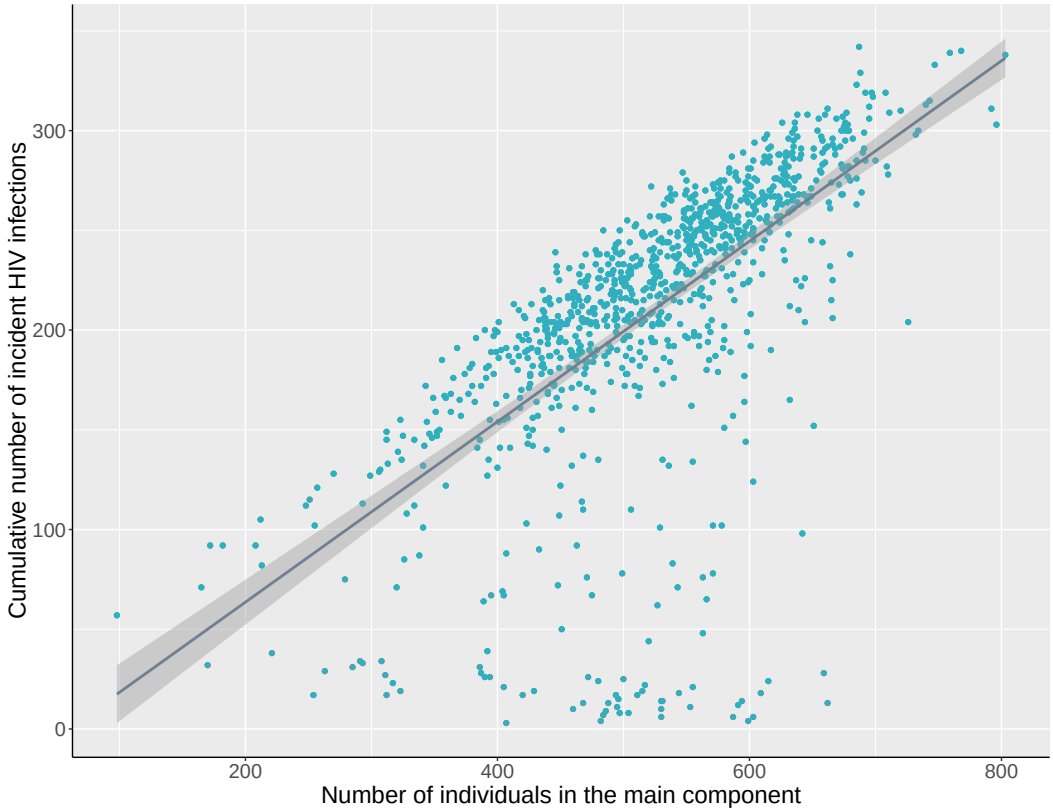


Figure 3: Size of the main component plotted against the number of cumulative incident HIV infections.

We make a first attempt to mitigate the effect of component size by constructing a different outcome: the prevalence of HIV in the main component at $t = 60$, the end of the simulation. We compare against this outcome to investigate the relationship between the network metrics and the intensity of the outbreak in the main component, controlled for the component's size. We report the correlation coefficients between each network metric and this new outcome.

Component Size Regression

Next, we construct a linear regression model to investigate the relationship between the main component size and the number of cumulative incident HIV infections. We use component size as the exposure and cumulative incident HIV infections as the outcome. The formal definition of a linear regression model follows:

$$Y_i = \beta' \mathbf{x} + \epsilon_i$$

where $\mathbf{x} \in \mathbb{R}^{10}$ is the vector of exposure variables, β' is the parameter vector, (also referred to as the regression coefficients), $\epsilon_i \in \mathbb{R}^{10}$ is the error term, and Y_i is the predicted outcome. The main inference and statistical analysis of a regression analysis focuses on the β values. In a linear regression, the coefficients are interpreted as the average effect in outcome per unit change in exposure, while controlling for other variable effects.

For this model, we write the following:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

Y_i = Predicted cumulative incident HIV infections (16)

X_i = Main component size (17)

This equation outputs a predicted number of cumulative incident infections, given an input of a simulated network's main component size.

We investigate the linear regression modelling assumptions using diagnostic plots. There is indeed a pattern to the residuals, violating the assumption of homoscedasticity (also known as the assumption of constant variance). This means that errors are not constantly distributed across the regression line. However, we allow this to occur as we intend to investigate the cause of this residuals pattern and believe these to be exceptions to our predicted rule.

Residuals Investigation

In a second attempt at investigating the second order effects, we calculate the component size regression models' residuals. The residuals of this network represent the distance between the number of cumulative incident infections in the simulations (the sample) and the predicted value output by our linear regression (i.e. the sample mean). We calculate a residual, r_i, by:

$$r_i = X_i - \bar{X}$$

X_i = Sample value (18)

$\bar{X}$ = Sample mean (19)

We are particularly interested in the residuals of this linear regression because the Fig. 3 reveals that there are cases that sit beneath the linear regression line, across the range of component sizes. These simulations correspond to higher residual values and fail to outbreak in the same manner as other networks with comparable main component size. We can plot these residuals against the other nine network metrics. We hypothesize that certain network characteristics cause networks to deviate from the predicted number of infections calculated using the main component size regression model. Specifically, we wish to determine if there are specific network characteristics that reduce the severity of the outbreak, controlling for main component size.

Further Analysis

Further investigation was done to determine second order effects, including the construction of two Poisson regression models. They are both included in Appendix Section 7.1.

4 Results

4.1 General Results

The model predicted 211 cumulative incident infections (95% SI: 17–306) over the five-year period within the total simulated population. This corresponds to an incidence rate of 0.18 infections (95% SI: 0.01–0.26) per 100 person years and an overall end-of-simulation prevalence of 0.96% (95% SI: 0.17–1.34%). The majority of infections occurred among PWID. Specifically, 177 cumulative incident infections (95% SI: 10–253) occurred among PWID, with an incidence rate of 12.2 infections (95% SI: 0.48–20.12) per 100 person years and an overall end-of-simulation prevalence of 43.74% (95% SI: 2.94–61.50%). Very few infections occurred among people who do not inject drugs. There were only 34 cumulative incident infections (95% SI: 7–58) within the non-injecting population. This corresponded to an incidence rate of 0.03 infections (95% SI: 0.006–0.05) per 100 person years and an overall end-of-simulation prevalence of 0.22% (0.10–0.33%). Fig. 4 below shows the cumulative incidence curves from each of the 1,000 simulations. The heterogeneity of outbreak severity is apparent in this figure.

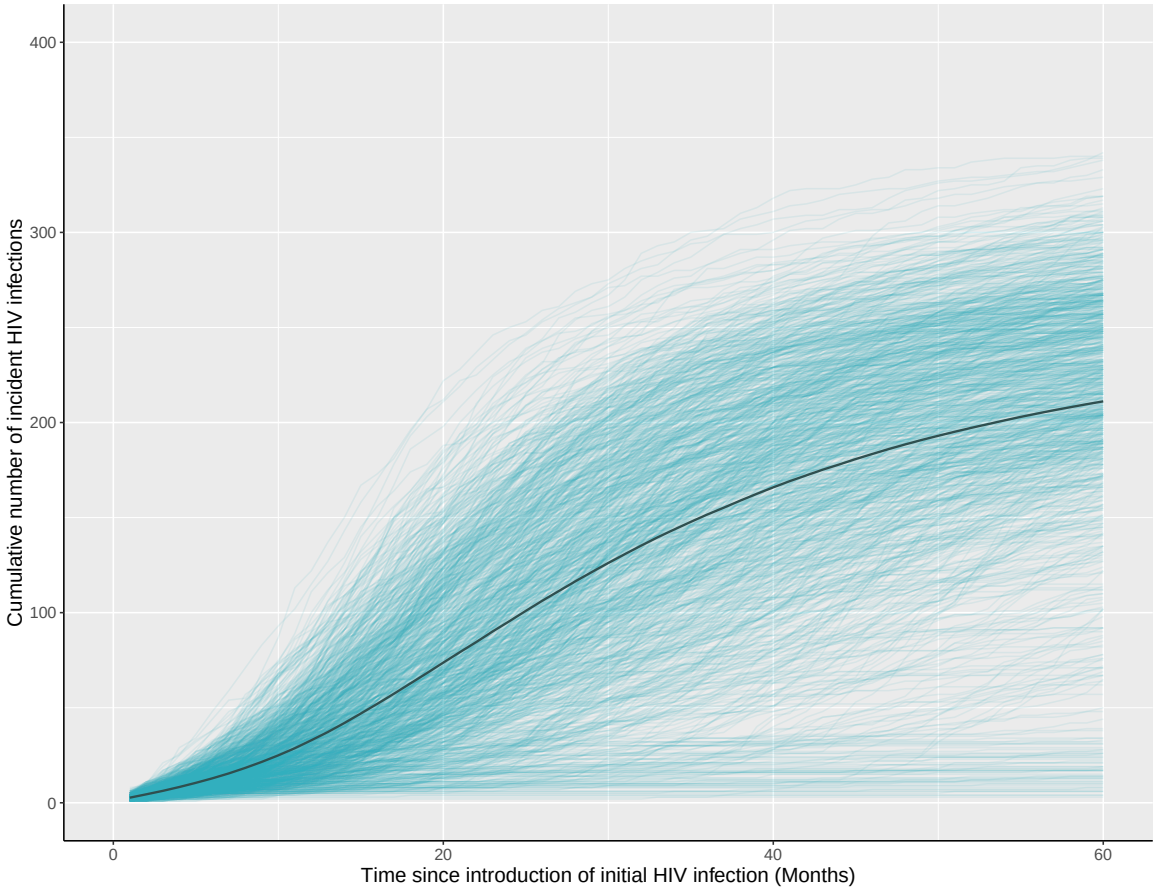


Figure 4: Incidence curves for each simulation with a mean line

We report a simulation interval (SI) for each of these measures rather than a confidence interval in an attempt to account for the stochastic framework of the model. The simulation intervals represent the 95% upper and lower limits of the aggregated simulation output. This approach is used in other modeling studies to explicitly address that these intervals are different from confidence intervals [33]. Because our "sample" is simply the number of times that we elect to run the simulation, we can narrow the "confidence interval" by running a greater number iterations. For this reason, we use simulation intervals in place of confidence intervals in this paper.

The model predicted the average degree of all HIV infected individuals at t=0 as 5 (95% SI: 0-9), the maximum degree out of all HIV infected individuals at t=0 as 6 (95% SI: 0-9), the average betweenness centrality in the main component as 0.0219 (95% SI: 0.0145-0.0366), the average random walk betweenness centrality in the main component as 0.0287 (95% SI: 0.0210-0.0425), the number of individuals in the main component as 526 (95% SI: 311-690), the density of the main component as 0.0041 (95% SI: 0.0030-0.007), the average geodesic distance among all points in the main component as 0.0874 (95% SI: 0.0674-0.104), the average centralization of all points in the main component as 0.0118 (95% SI: 0.0076-0.0175), the transitivity in the main component as 0.0013 (95% SI: 0-0.0069), and the proportion of individuals involved in 2-cores in the main component as 0.1510 (95% SI: 0.0667-0.2171).

4.2 Correlation Coefficient Analysis

We report on the coefficient values for both outcomes. First with the primary outcome, the number of cumulative incident HIV infections, and second with the secondary outcome, the prevalence of HIV in the main component at the end of the simulation (i.e. $t = 60$). We also report 95% simulation intervals for each of the coefficients that we aim to estimate.

Network Metric	Cumulative Incident Infections	Prevalence in Main Component
Avg degree of HIV infected individuals	0.391 (0.337, 0.442)	0.481 (0.432, 0.527)
Max degree of HIV infected individuals	0.452 (0.402, 0.500)	0.535 (0.489, 0.578)
Betweenness centrality	-0.669 (-0.702, -0.633)	-0.249 (-0.307, -0.190)
Random walk betweenness centrality	-0.672 (-0.705, -0.636)	-0.239 (-0.297, -0.180)
Main component size	0.648 (0.611, 0.683)	0.107 (0.045, 0.168)
Main component density	-0.546 (-0.588, -0.501)	-0.063 (-0.125, -0.001)
Geodesic distance	0.251 (0.192, 0.309)	0.349 (0.293, 0.402)
Centralization	-0.402 (-0.453, -0.349)	0.014 (-0.048, 0.076)
Transitivity	-0.004 (-0.066, 0.0585)	0.029 (-0.033, 0.091)
Proportion of 2-Cores	0.431 (0.379, 0.480)	0.223 (0.163, 0.281)

This tells us that we predict a positive association between average degree of HIV infected individuals at $t = 0$ (Cor: 0.39, 95% SI: 0.337, 0.442), maximum degree of HIV infected individuals at $t = 0$ (Cor: 0.452, 95% SI: 0.402, 0.500), main component size (Cor: 0.648, 95% SI: 0.611, 0.683), average geodesic distance in the main component (Cor: 0.251, 95% SI: 0.192, 0.309), and proportion of the individuals located 2-cores in the main component (Cor: 0.431, 95% SI:

0.379, 0.480) and the number of cumulative incident infections.

We predict a negative association between average betweenness centrality in the main component (Cor: 0.669, 95% SI: -0.702, -0.633), average random walk betweenness centrality in the main component (Cor: -0.672, 95% SI: -0.705, -0.636), main component density (Cor: -0.546, 95% SI: -0.588, -0.501), average centralization in the main component (Cor: -0.402, 95% SI: -0.453, -0.349), and transitivity in the main component (Cor: -0.004, 95% SI: -0.066, 0.0585) and the number of cumulative incident infections.

This also tells us that we predict a positive association between average degree of HIV infected individuals at $t = 0$ (Cor: 0.481, 95% SI: 0.432, 0.527), maximum degree of HIV infected individuals at $t = 0$ (Cor: 0.535, 95% SI: 0.489, 0.578), main component size (Cor: 0.107, 95% SI: 0.045, 0.168), average geodesic distance in the main component (Cor: 0.349, 95% SI: 0.293, 0.402), average centralization in the main component (Cor: 0.014, 95% SI: -0.048, 0.076), transitivity in the main component (Cor: 0.029, 95% SI: -0.033, 0.091), and proportion of the individuals located 2-cores in the main component (Cor: 0.223, 95% SI: 0.163, 0.281) and the final prevalence in the main connected component.

We predict a negative association between average betweenness centrality in the main component (Cor: -0.249, 95% SI: -0.307, -0.190), average random walk betweenness centrality in the main component (Cor: -0.239, 95% SI: -0.297, -0.180), and main component density (Cor: -0.063, 95% SI: -0.125, -0.001) and the final prevalence in the main connected component.

The majority of the exposures retain the same association across the change of outcome. However, centralization and transitivity are negatively associated with the number of incident infections, but positively associated with the final prevalence in the main connected component. Note that the 95% SI for transitivity falls both positively and negatively for both outcomes, rendering the result statistically insignificant. Also, note that 95% SI for centralization remains negative for both cumulative incidence, implying that result statistically significant, while falling both positive and negative for prevalence in the main connected component, a statistically insignificant result. We conclude that changes in both of these metrics are insubstantial.

4.3 Component Size Regression

The linear regression model of component size with incident infections produced a regression coefficient of 0.4524 (p-value: $<< 0.001$). This tells us that with every unit increase in component size, we predict that the number of incident infections will increase by 0.4524. This plot is included in Fig. 5 below with the predicted regression line. The updated coloring indicates if the spontaneous seroconversion occurred in the main component or not. 40 of the 1000 simulations are instances where infection did not breach the main component. These 40 instances are colored in red, while the other simulations remain in blue. We identified these instances by measuring the degree of the HIV-infected nodes at $t = 0$. If there were no HIV-infected nodes to be measured, the network analysis script outputs 0. All iterations with degree outputs of zero were therefore identified as iterations without HIV in the main component. Plots will be colored in this manner for the remainder of the paper.

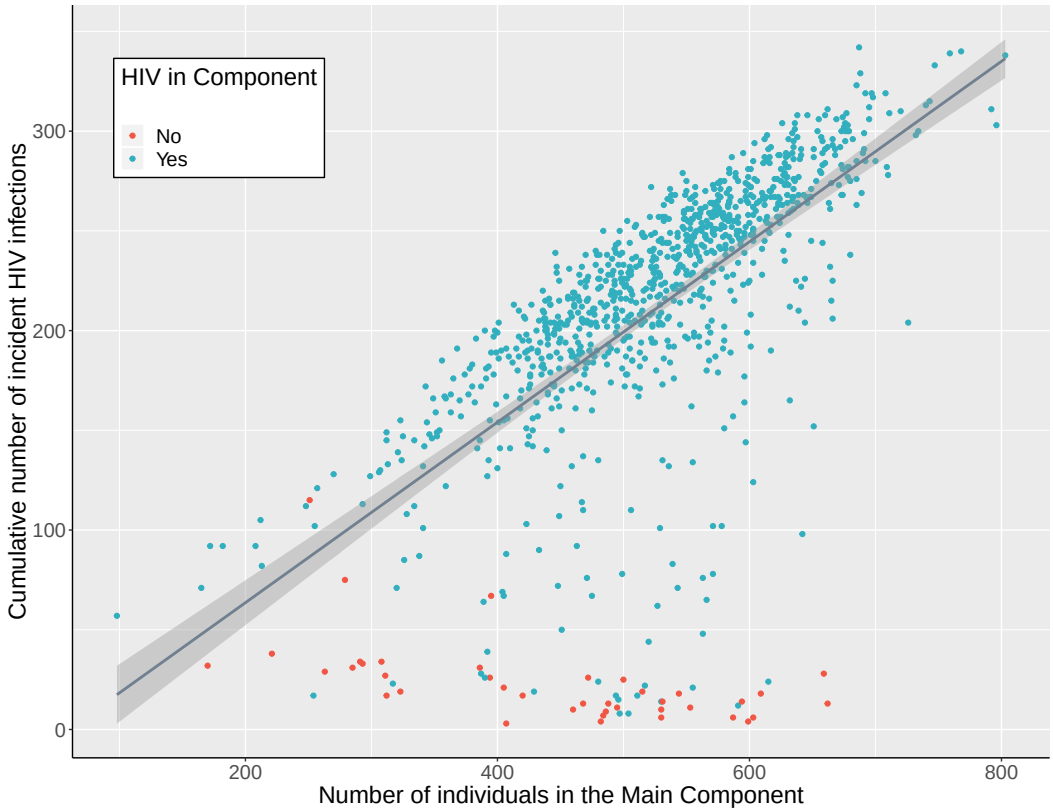


Figure 5: Size of the main component plotted against the number of cumulative incident HIV infections colored by whether or not HIV infection was present the main component.

4.4 Residuals Investigation

The range of residuals from the component size regression model is (-259.59, 64.13), with a quartile range of (-4.75, 28.86). Figure 6 below is a visual representation of the spread of these residual values. The blue and red groupings represent the same as above: indicating whether or not HIV was introduced into the main component during network formation.

We note that 104 of the simulations fall substantially below the regression line. We classify a substantial deviation from the regression line as being at or lower than -50 units (incident infections) away from their predicted value. This is represented by the black, dotted, vertical line laid over the box plot. This classification lines up closely with the separation of outliers.

The simulations without HIV infection in their main component are represented in a separate grouping in Fig. 6 to explicitly address that they should not be considered in our following outlier analysis. These runs have strong deviation off the regression line because of their lack of HIV in the main component, not because of our hypothesized second order network structure effect. Therefore, we are concerned only with the outliers shown in the first boxplot. These amount to 64 simulations.

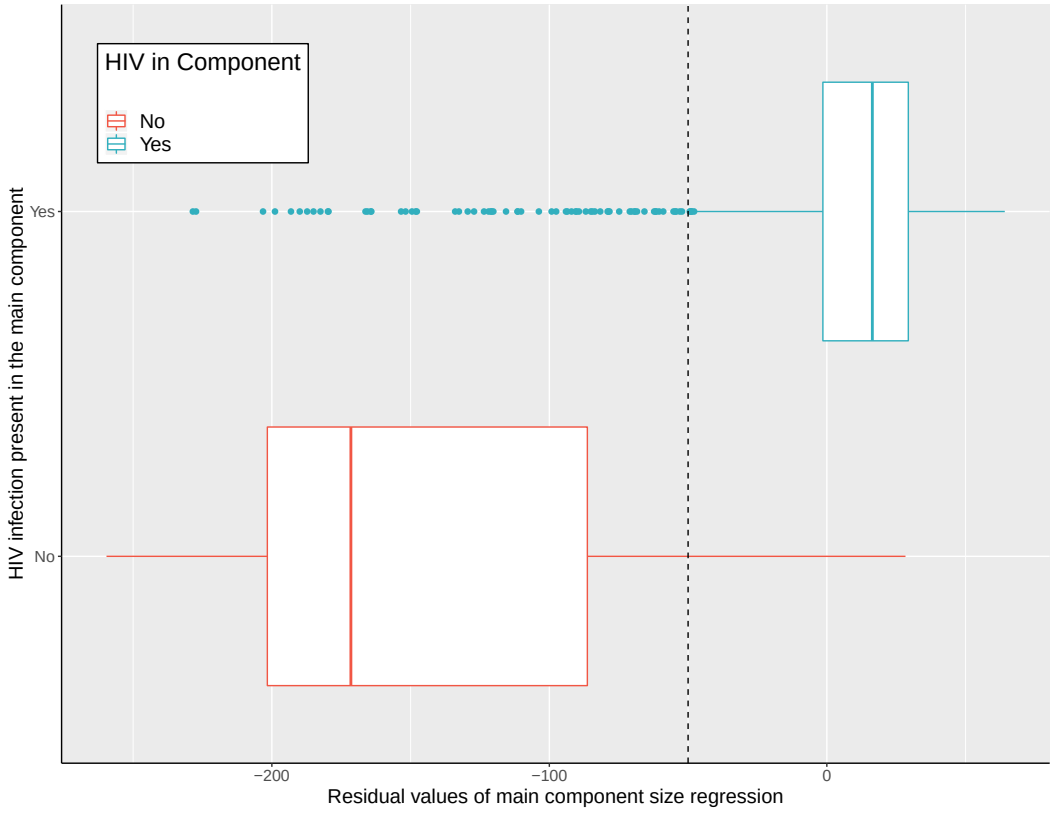


Figure 6: Spread of residual values of main component size regression model

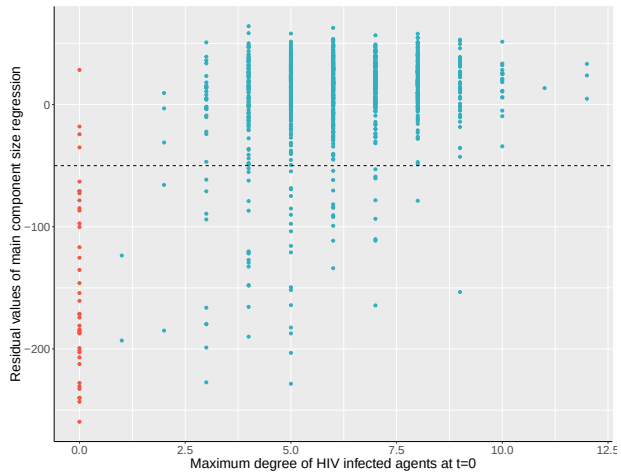
Figure 7 below presents a subset of the plots of each metric graphed against the residuals of our linear regression component size model. The full set of plots is included in Appendix Section 7.3. The coloring is again included along with the dotted line representing substantial deviation off of the regression line.

We do not have substantial results to report from any of the residuals plots. If the deviation from the component size regression line corresponded to specific network structures, we would expect to see a trend in these plots. Specifically, we would expect the group of points with Y-values lower than -50 (meaning that they are falling substantially below the regression line) deviating from the spread of the others. We do not see this pattern emerge in a clear way in any of our nine graphs (we exclude the comparison with component size).

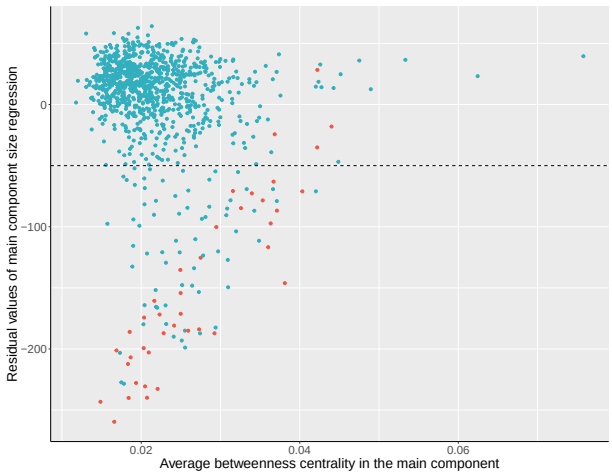
Again, we recall the need to de-emphasize any pattern or effect of the red colored points–as these are not points that pertain to this outlier analysis.

The only mild patterns to make note of are that the iterations with large residuals appear to have slightly lower maximum degree of HIV infected individuals at $t = 0$ (Fig.7a), lower main component density (Fig.7c), and lower transitivity (Fig.7f). All three of these graphs show the runs with small residuals as appearing to spread farther right on the graph, while the high residuals runs are more closely retained to lower values of each of the metrics. Notably, these are reported observations from the scatter plots, not from statistical analysis.

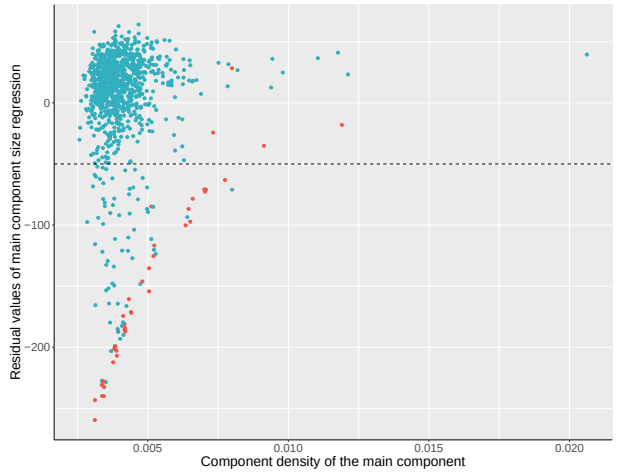
These do indeed all build on our intuition: a lower degree of HIV infected individuals at $t = 0$ prevents a swift beginning of disease transmission; a less dense network is one with less edges (i.e. fewer paths for the disease to maneuver); and a lower amount of transitivity indicates lower complete triangles, and therefore less opportunity for dense, high risk sections of the network. We come to a wary conclusion that lower degree of HIV infected individuals at $t = 0$, lower density, and lower transitivity may correspond with less severe HIV outbreak behavior (at least as it compares to our prediction based on main component size).



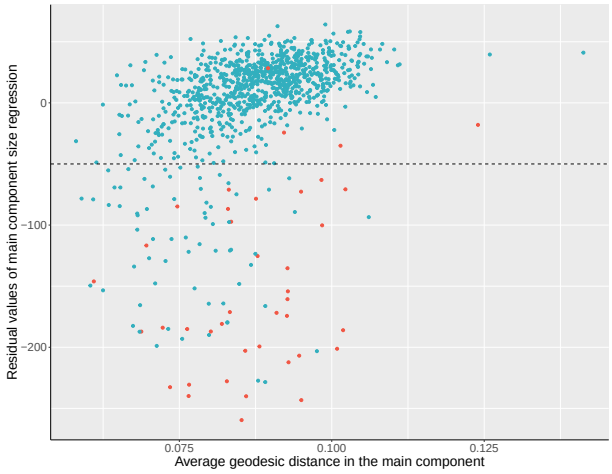
(a) Max degree of HIV infected agents



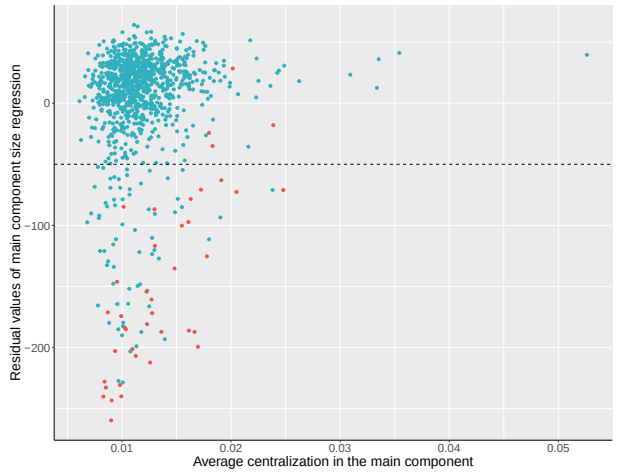
(b) Average betweenness centrality



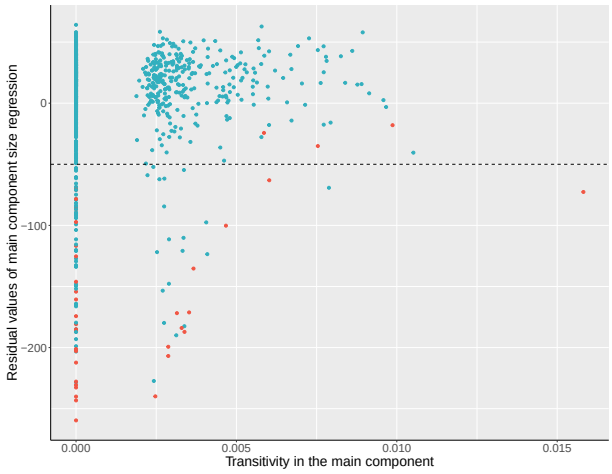
(c) Main component density



(d) Average geodesic distance



(e) Average centralization



(f) Transitivity

Figure 7: Network metrics graphed against the residuals of our main component size linear regression model.

5 Discussion

5.1 Summary and Context

Our main finding from this analysis is the strong, first order effect of main component size on the number of incident infections. As the number of individuals connected to the main component grew, so did the number of cumulative incident infections at the end of the five-year simulation period. Intuitively, we recognize that as we increase the upper bound in the number of susceptible PWID in a component infected with HIV, the infection is able to spread further. As mentioned in Section 2.4, large main components are common among PWID communities, increasing the risk of rapid HIV spread upon the virus' entry into the population.

Further investigation into the second order effects of network structure, through the investigation of the outliers of our component size regression model, revealed that decreases in the maximum degree of HIV infected agents at $t = 0$, main component density, and transitivity in the main component might also result in less extreme epidemic behavior. However, we note that the support we have for this conclusion is both minimal and observational.

These conclusions align with many of the articles and analyses discussion within Section 3.2. This analysis reinforces that injection drug use networks are particularly at risk for rapid HIV transmission should a member of the network contract an HIV infection. Common characteristics of IDU networks, such as a large main component, relatively high degree distributions, high density, and presence of high amounts of transitivity align with extreme HIV epidemic behavior. These characteristics leave injection-drug use networks at a particularly high-risk for infectious disease acquisition. Implementing intervention strategies that directly address the unique characteristics of IDU networks will be imperative in supporting the health of these communities.

5.2 Limitations

The most notable limitation to this work is the potential bias regarding the input distributions used to build the simulated networks. The degree distributions used to construct our 1,000 models were provided by the CDC and the work done by Ells Campbell [9]. Importantly, when the Indiana State Department of Health (ISDH) and CDC conducted the contact tracing investigation, they halted their tracing when they reached an HIV-negative person. Therefore, we are missing connections between HIV-negative individuals and other HIV-negative individuals in the network. We recognize that this leaves the degree distributions incomplete, and almost certainly underestimates the true degree distributions of the Scott County Network.

Other limitations to this study also require discussion. We did not model heterogeneity in injection behavior depending on the drug of choice. Although 92% of all diagnosed PWID in Scott County reported use of oxymorphone, other substances present in the community inform different injection behavior [15]. With different choices in drug use comes the possibility

of differences in number of injection events and differences in preparation procedures. The model also does not account for the increasing size of the PWID population and its influence on injection behavior [3, 32]. The model also does not account for heterogeneity in injection behavior. Injection behavior can depend on one's location within the network (i.e. if one is located within a 2-core, within the main connected component, or a member of a small component) [4]. Lastly, observed data from the setting of Scott County was used to parameterize the model wherever possible. However, as with most agent-based models, multiple other sources were used to parameterize the model where data specific to Scott County were unavailable. This introduces bias into the model and influences the model's external validity. We believe that the magnitude of this bias will remain minimal as the vast majority of the parameters were derived from Scott County itself.

5.3 Future Work

Opportunities for future work are abundant. What follows is a report on the future steps we are contemplating taking over the next few months.

Fix Component Size

In another attempt to control for the effect of component size, we can form a new set of networks where each of the main components are comparable in size to the observed main component. We could then compare the other network metrics across this new set of networks. This would allow us the opportunity to analyze other structural characteristics, independent of component size. We would implement this by allowing TITAN to build a network, measure the main component size, and add it to our network set if the component falls within a predetermined range. If it does not, we discard it and repeat the process until our set has 1,000 elements.

Heterogeneous Injection Behavior

As mentioned in the Section 5.2, injection behavior (such as the frequency of injection acts) can depend on one's location within the network. These two characteristics are currently independently assigned to individuals during population formation. We could assign a scaling factor to the target number of injection acts assigned to an agent depending on their location within the network. This would blur the original independence assumption and act as a form of sensitivity analysis.

Measure Initial Infection Characteristics

The confirmation that the degree of HIV infected agents at $t = 0$ is positively associated with an increase in incident infections encourages further investigation into the effects of initial infection characteristics. Whether the initial infection is located in a 2-core or a dense part of the network would reveal further affects of the location of the initial infection. Further, we may wish to measure the betweenness centrality and random walk betweenness centrality of just the initial infection. This investigation could address the question of how HIV enters into injection-drug use networks. We can alter the original network analysis script to measure any

number of characteristic of the HIV infected agents at $t = 0$.

Variability and Noise

We propose running multiple iterations of each of the 1,000 simulated networks and spontaneous seroconversions. Essentially, we isolate the noise "inside" TITAN, and allow TITAN to run the same infection within the same network multiple times for each of the 1,000 networks formed. We suggest 12 repetitions with the aim to balance computation time and while still reducing variation in the samples. This will allow us to investigate the variability possible from each of our 1,000 runs. We hypothesize that different amounts of variability will be associated with different network metrics. Network edge cases (i.e. networks with specific qualities) might experience very different levels of outbreak severity, whereas average networks might have much more of a predetermined size of outbreak. We can implement this with the addition of another PRNG for the HIV infection introduction.

Network-Based Interventions

Lastly, we could build off this paper's investigation of IDU network structure to add network-based interventions into the model. We mentioned a few network-based interventions in Section 2.2, but an in-depth literature would be needed to assess if any of those interventions (or others) could be implemented into TITAN. If so, this would be an ideal opportunity to investigate the development of intervention strategies that specifically address the uniqueness of IDU network structure.

6 References

1. Cdc.gov. (2019). Injection Drug Use — HIV Risk and Prevention — HIV/AIDS — CDC. [online] Available at: https://www.cdc.gov/hiv/risk/idu.html [Accessed 20 Mar. 2019].

2. Degenhardt L, Peacock A, Colledge S, Leung J, Grebely J, Vickerman P, Stone J, Cunningham EB, Trickey A, Dumchev K, Lynskey M. Global prevalence of injecting drug use and sociodemographic characteristics and prevalence of HIV, HBV, and HCV in people who inject drugs: a multistage systematic review. The Lancet Global Health. 2017; 5(12):e1192-207.

3. Des Jarlais DC, Kerr T, Carrieri P, Feelemyer J, Arasteh K. HIV infection among persons who inject drugs: ending old epidemics and addressing new outbreaks. AIDS (London, England). 2016; 30(6):815.

4. Friedman SR, Neaigus A, Jose B, Curtis R, Goldstein M, Ildefonso G, Rothenberg RB, Des Jarlais DC. Sociometric risk networks and risk for HIV infection. American journal of public health. 1997; 87(8):1289-96.

5. Estrada, Ernesto. The Structure of Complex Networks: Theory and Applications. Oxford University Press, 2016.

6. Shirley MD, Rushton SP. The impacts of network topology on disease spread. Ecological Complexity. 2005; 2(3):287-99.

7. Keeling MJ, Eames KT. Networks and epidemic models. Journal of the Royal Society Interface. 2005; 2(4):295-307.

8. Keeling M. The implications of network structure for epidemic dynamics. Theoretical population biology. 2005; 67(1):1-8.

9. Campbell EM, Jia H, Shankar A, Hanson D, Luo W, Masciotra S, Owen SM, Oster AM, Galang RR, Spiller MW, Blosser SJ. Detailed transmission network analysis of a large opiate-driven outbreak of HIV infection in the United States. The Journal of infectious diseases. 2017; 216(9):1053-62.

10. Young AM, Jonas AB, Mullins UL, Halgin DS, Havens JR. Network structure and the risk for HIV transmission among rural drug users. AIDS and Behavior. 2013; 17(7):2341-51.

11. Friedman SR, Neaigus A, Jose B, Curtis R, Des Jarlais D. Networks and HIV risk: an introduction to social network analysis for harm reductionists. International Journal of Drug Policy. 1998; 9(6):461-9.

12. De P, Cox J, Boivin JF, Platt RW, Jolly AM. The importance of social networks in their association to drug equipment sharing among injection drug users: a review. Addiction. 2007; 102(11):1730-9.

13. Dombrowski K, Khan B, Habecker P, Hagan H, Friedman SR, Saad M. The interaction of risk network structures and virus natural history in the non-spreading of HIV among people who inject drugs in the early stages of the epidemic. AIDS and Behavior. 2017; 21(4):1004-15.

14. Khan B, Dombrowski K, Saad M, McLean K, Friedman S. Network firewall dynamics and the subsaturation stabilization of HIV. Discrete dynamics in nature and society. 2013; 2013.

15. Peters PJ, Pontones P, Hoover KW, Patel MR, Galang RR, Shields J, Blosser SJ, Spiller MW, Combs B, Switzer WM, Conrad C. HIV infection linked to injection use of oxymorphone in Indiana, 2014–2015. New England Journal of Medicine. 2016; 375(3):229-39.

16. Conrad C, Bradley HM, Broz D, Buddha S, Chapman EL, Galang RR, Hillman D, Hon J, Hoover KW, Patel MR, Perez A. Community outbreak of HIV infection linked to injection drug use of oxymorphone—Indiana, 2015. MMWR. Morbidity and mortality weekly report. 2015 May 1;64(16):443.

17. Tanner M, Halleck B, Dasgupta S, et al. Utilization of HIV testing and prevention services among people who inject drugs - Indiana, 2016. Conference on Retroviruses and Opportunistic Infections. Boston, Massachusetts, 2018.

18. Doyle JS, Degenhardt L, Pedrana AE, et al. Effects of HIV antiretroviral therapy on sexual and injecting risk-taking behavior: A systematic review and meta-analysis. Clin Infect Dis 2014; 59(10): 1483-94.

19. Crosby RA, Oser CB, Leukefeld CG, Havens JR, Young A. Prevalence of HIV and risky sexual behaviors among rural drug users: Does age matter? Ann Epidemiol 2012; 22(11): 778-82.

20. Patel P, Borkowf CB, Brooks JT, Lasry A, Lansky A, Mermin J. Estimating per-act HIV transmission risk: A systematic review. AIDS 2014; 28(10): 1509-19.

21. Bellan SE, Dushoff J, Galvani AP, Meyers LA. Reassessment of HIV-1 acute phase infectivity: Accounting for heterogeneity and study design with simulated cohorts. PLoS Med 2015; 12(3): e1001801.

22. Sterne JA, Hernan MA, Ledergerber B, et al. Long-term effectiveness of potent antiretroviral therapy in preventing AIDS and death: A prospective cohort study. Lancet 2005; 366(9483): 378- 84.

23. Henderson ER, Subramaniam DS, Chen J. Rural-urban differences in HIV testing among US Adults: Findings from the Behavioral Risk Factor Surveillance System. Sex Transm Dis 2018; 45(12): 808-12.

24. Karch DL, Gray KM, Shi J, Hall HI. HIV Infection care and viral suppression among people who inject drugs, 28 U.S. jurisdictions, 2012-2013. Open AIDS J 2016; 10: 127-35.

25. Skarbinski J, Rosenberg E, Paz-Bailey G, et al. Human immunodeficiency virus transmission at each step of the care continuum in the United States. JAMA Intern Med 2015; 175(4): 588-96.

26. Newman ME. A measure of betweenness centrality based on random walks. Social networks. 2005; 27(1):39-54.

27. Estrada E, Higham DJ, Hatano N. Communicability betweenness in complex networks. Physica A: Statistical Mechanics and its Applications. 2009 Mar 1;388(5):764-74.

28. Tarwater PM, Martin CF. Effects of population density on the spread of disease. Complexity. 2001; 6(6):29-36.

29. Seidman SB. Network structure and minimum degree. Social networks. 1983; 5(3):269-87.

30. Wasserman S, Faust K. Social network analysis: Methods and applications. Cambridge university press; 1994 Nov 25.

31. Diestel, R. (2010). Graph Theory (5th Edition). Berlin: Springer.

32. Suh T, Mandell W, Latkin C, Kim J. Social network characteristics and injecting HIV-risk behaviors among street injection drug users. Drug and alcohol dependence. 1997; 47(2):137-43.

33. Jenness SM, Goodreau SM, Rosenberg E, Beylerian EN, Hoover KW, Smith DK, et al. Impact of the Centers for Disease Control's HIV Preexposure Prophylaxis Guidelines for Men Who Have Sex With Men in the United States. J Infect Dis. 2016; 214(12):1800-7.

34. United States Census Bureau. American Community Survey, 2006-2010. Washington, District of Columbia: United States Department of Commerce, 2011.

35. Indiana State Department of Health. Spotlight on HIV/STD/Viral Hepatitis: Indiana Semi-Annual Report. Indianapolis, Indiana: State of Indiana, 2010.

36. National Center for Health Statistics. Multiple Cause of Death, 1999-2016. Hyattsville, Maryland: Centers for Disease Control and Prevention; 2018.

37. Young AM, Halgin DS, Havens JR. Relationship-level analysis of drug users' anticipated changes in risk behavior following HIV vaccination. AIDS Care 2015; 27(8): 1000-4.

7 Appendices

7.1 Poisson Regression Models

We attempt to employ a form of logistic regression rather than a linear regression. As previously mentioned in Section 3.3, the diagnostic plots revealed that the residuals do not fall into a linear trend. This encourages our use of logistic regression as opposed to a basic linear regression model.

We make use of Poisson regression as our generalized linear model because our outcome of incident infections is a count. We make two major assumptions. First, we assume that the number of incident infections, our outcome, can be represented through a Poisson distribution. Second, we assume that the logarithm of this outcome's expected value can be represented through a linear combination of the network metrics–the exposure variables. The definition of our model follows:

$$\log(\mathrm{E}[Y_i|\mathbf{x}]) = \beta'\mathbf{x} + \epsilon_i$$

where $\mathbf{x} \in \mathbb{R}^{10}$ is the vector of exposure variables, β' is the parameter vector, (also referred to as the regression coefficients), $\epsilon_i \in \mathbb{R}^{10}$ is the error term, and Y_i is the predicted outcome. In a Poisson regression, the coefficients are interpreted as the average effect in outcome per unit change in log odds of exposure, while controlling for other variable effects. In other worlds, the relative risk or risk ratios.

Lastly, we note that we need to exponentiate to retrieve easily interpreted incidence rate ratios. We report on 95% simulation intervals for each of the coefficients that we aim to estimate. We exponentiate the coefficients to maximize interpretability, and we find the confidence intervals for e^{β}. We calculate these confidence intervals using the following equation:

$$e^{\hat{\beta} \pm 1.96\mathrm{SE}(\hat{\beta})}$$

Complete Model

The following is a multivariable analysis of structural network characteristics associated with the number of cumulative incident HIV infections at the end of a five-year simulation period. This first model includes all network statistics measured in the network analysis and will therefore be referred to as the "complete model."

We define the complete model as follows:

$$\log(\mathrm{E}[Y_i|\mathbf{x}]) = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \beta_3 X_{3,i} + \beta_4 X_{4,i} + \beta_5 X_{5,i} + \beta_6 X_{6,i} + \beta_7 X_{7,i} + \beta_8 X_{8,i} + \beta_9 X_{9,i} + \beta_{10} X_{10,i} + \epsilon_i$$

Y_i = Predicted cumulative incident HIV infections (20)

$X_{1,i}$ = Average degree of HIV infected individuals at $t = 0$ (21)

$X_{2,i}$ = Maximum degree of HIV infected individuals at $t = 0$ (22)

$X_{3,i}$ = Average betweenness centrality in main component (23)

$X_{4,i}$ = Average random walk betweenness centrality in main component (24)

$X_{5,i}$ = Main component size (25)

$X_{6,i}$ = Main component density (26)

$X_{7,i}$ = Average geodesic distance in main component (27)

$X_{8,i}$ = Average centralization in main component (28)

$X_{9,i}$ = Transitivity in main component (29)

$X_{10,i}$ = Proportion of individuals located 2-cores in the main component (30)

These reported coefficients characterize the association of the exposure and the difference in log odds of the number of incident infections, while keeping all other predictors the same. For each metric, if the reported coefficient is above 1, the number of incident infections increases multiplicatively by the coefficient amount for every unit increase in the network metric, while holding all other metrics constant. On the other hand, if the reported coefficient is below 1, the number of incident infections decreases multiplicatively by the coefficient amount for every unit increase in the network metric, while holding all other metrics constant.

Network Metric	Incidence Rate Ratio (95% SI)	P value
Avg degree of HIV infected individuals	1.004 (0.999-1.009)	0.0631
Max degree of HIV infected individuals	1.068 (1.063-1.073)	<0.001
Avg betweenness centrality	6.154e-10 (3.897e-13-9.572e-07)	<0.001
Avg random walk betweenness centrality	4.119e-17 (2.456e-20-6.948e-14)	<0.001
Main component size	1.0003 (1.0002-1.0005)	<0.001
Main component density	4.347e+73 (3.511e+58-4.419e+88)	<0.001
Avg geodesic distance	4.498e-5 (4.070e-6-4.964e-4)	<0.001
Avg centralization	5.402e-7 (4.430e-8-6.567e-6)	<0.001
Transitivity	0.326 (0.037-2.869)	0.3131
Proportion of 2-cores	5.574 (4.212-7.374)	<0.001

Iterations with higher numbers of cumulative incident infections had a larger maximum degree of HIV infected individuals at $t = 0$ (IRR: 1.068, 95% SI: 1.063, 1.073), main component size (IRR: 1.0003, 95% SI: 1.0002, 1.0005), main component density (IRR: 4.347e+73, 95% SI: 3.511e+58, 4.419e+88), and proportion of individuals located 2-cores in the main component (IRR: 5.574, 95% SI: 4.212, 7.374). All of our other network measurements were not independently associated with an increase in cumulative incident infections, including average betweenness centrality in the main component (IRR: 6.154e-10, 95% SI: 3.897e-13, 9.572e-07),

random walk betweenness centrality in the main component (IRR: 4.119e-17, 95% SI: 2.456e-20, 6.948e-14), average geodesic distance in the main component (IRR: 4.498e-5, 95% SI: 4.070e-6, 4.964e-4), and average centralization in the main component (IRR: 5.402e-7, 95% SI: 4.430e-8, 6.567e-6). Neither the average degree of HIV infected individuals at $t = 0$ nor the transitivity in the main component produced significant results.

However, the estimates in the complete model being so large and so small indicate a possible collinearity between covariates. For this reason, we decide to build a reduced model, where specific exposure variables are removed from the model in an attempt to reduce collinearity.

Reduced Model

Using the Variance Inflation Factors (vif function in R), we decided to remove average degree, random walk betweenness centrality, main component density, geodesic distance, and centralization. The vif function calculates variance-inflation and generalized variance-inflation factors for linear, generalized linear, and other models. We reduce the model until each of the exposure variable remaining have low vif values. We retain only max degree, main component size, transitivity, and proportion of 2-cores. We define the reduced model as follows:

$$\log(\mathrm{E}[Y_i|\mathbf{x}]) = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \beta_3 X_{3,i} + \beta_4 X_{4,i} + \epsilon_i$$

Y_i = Predicted cumulative incident HIV infections (31)

$X_{1,i}$ = Maximum degree of HIV infected individuals at $t = 0$ (32)

$X_{2,i}$ = Main component size (33)

$X_{3,i}$ = Transitivity in main connected component (34)

$X_{4,i}$ = Proportion of individuals located 2-cores in the main component (35)

The output of the reduced model is as follows. The interpretation of the coefficients is the same as described above in reference to the complete model.

Network Metric	Incidence Rate Ratio (95% SI)	P value
Max degree of HIV infected individuals	1.073 (1.070-1.075)	<0.001
Main component size	1.0019 (1.0018-1.0019)	<0.001
Transitivity	5.574 (0.1309-9.695)	<0.001
Proportion of 2-cores	6.362 (5.585-7.247)	0.912

Iterations with higher numbers of cumulative incident infections had a larger maximum degree of HIV infected individuals at $t = 0$ (IRR: 1.073, 95% SI: 1.070-1.075), main component size (IRR: 1.0019, 95% SI: 1.0018-1.0019), transitivity in the main component (IRR: 5.574, 95% SI: 0.1309-9.695), and proportion of 2-cores (IRR: 6.362, 95% SI: 5.585-7.247). Note that each of these findings is significant outside of the proportion of the individuals located 2-cores in the main component.

Likelihood Ratio Test

We employ a Likelihood Ratio Test of Nested Models (LRT) to formalize the comparison between the two models. We consider these nested models because all of the variables in the reduced model are included in the complete model. The LRT tests whether content is significantly reduced relative to reduction in number of parameters. The test statistic is twice the difference in the maximum log-likelihoods and is compared with a Chi-squared distribution. Formally, we define the test statistic as:

LR Test Statistic $= -2$ (max likelihood for reduced model$-$max likelihood for complete model)

$$\text{LR Test Statistic} = -2log\ \frac{\text{max likelihood for reduced model}}{\text{max likelihood for complete model}}$$

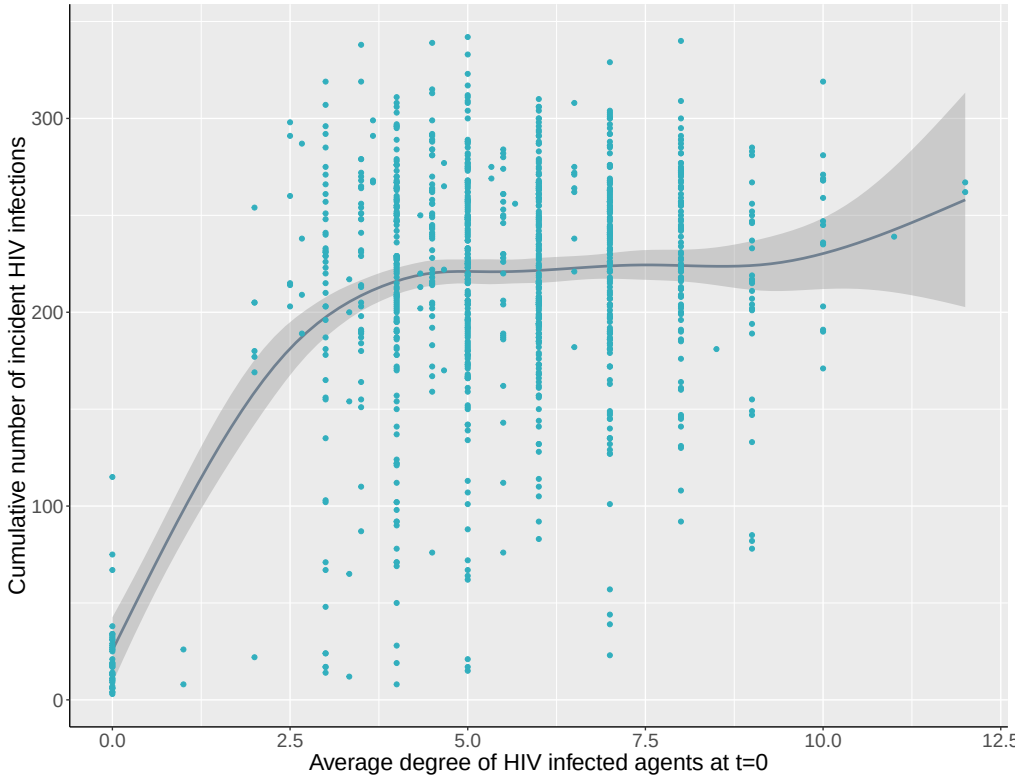
The output of the test is as follows:

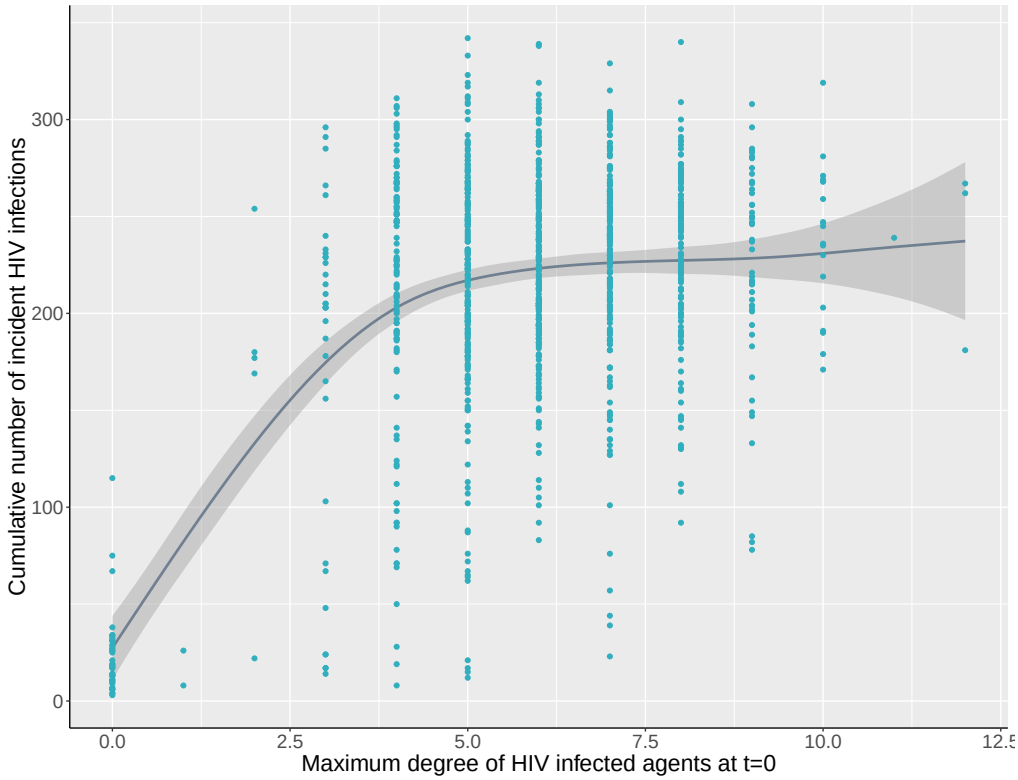
Poisson Model	Degrees of Freedom	Log Likelihood
Complete Model	5	-11268
Reduced Model	11	-10736

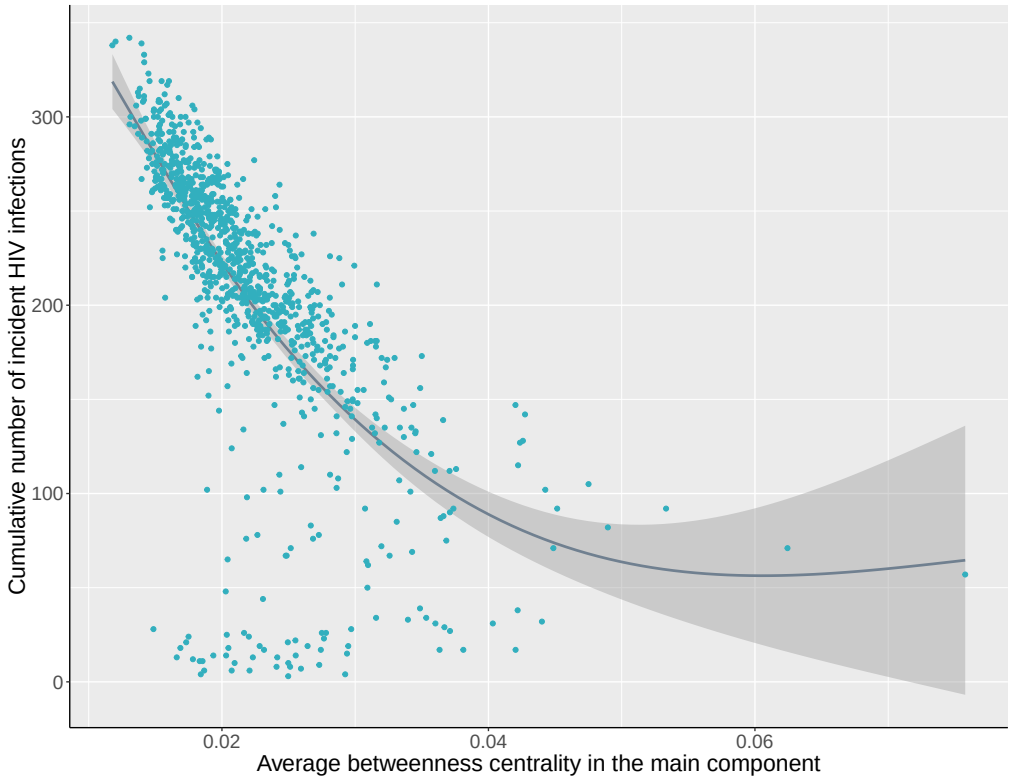
When we compare twice the difference in the log-likelihoods with a Chi-squared distribution, we determine a p-value of < 0.001. This indicates that the reduced model is indeed a more appropriate predictive tool than the complete model.

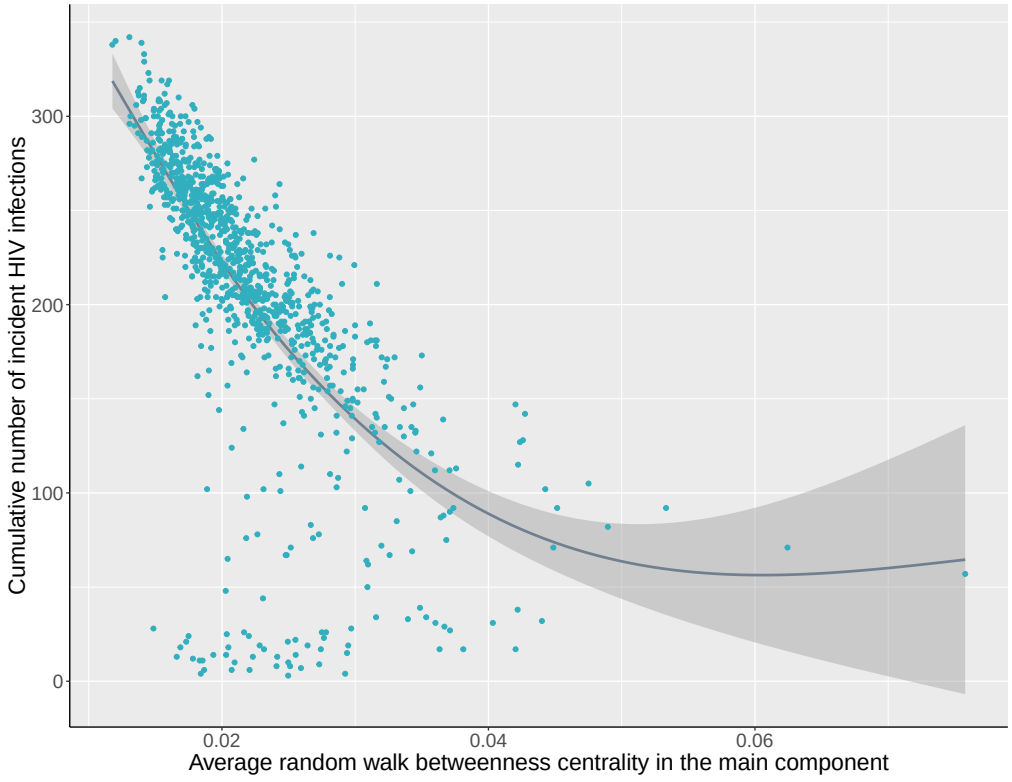
7.2 Primary Analysis Plots

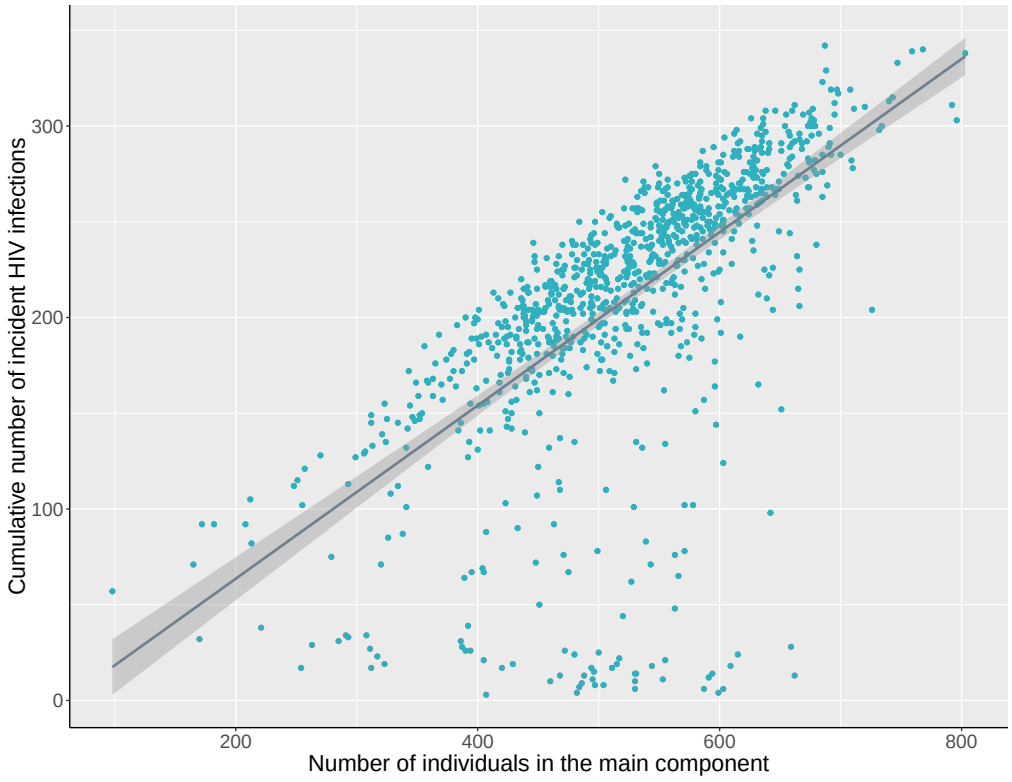
Section 7.2 provides plots of each network metric against the cumulative number of incident infections at the end of the five-year simulation. Each plot includes a smoothing function graphed in grey (generated using the basic stat-smooth function in R). Note this is only a visual-aid, not a formal analysis tool.

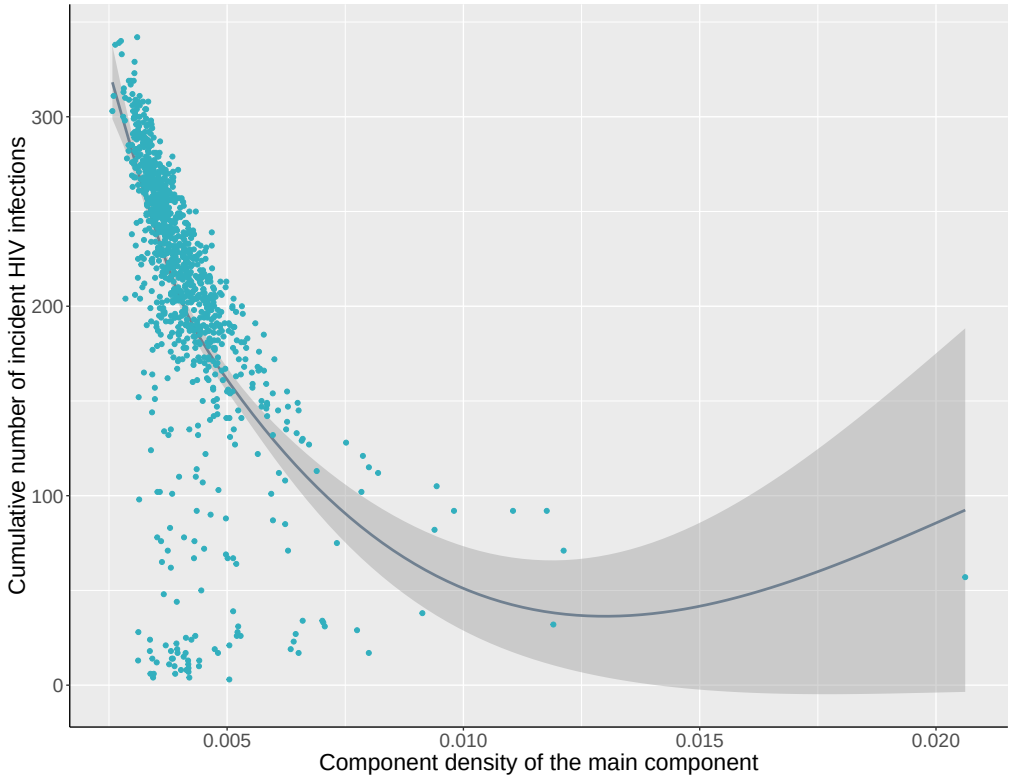

Cumulative number of incident HIV infections
300
200
100
0
0.0
2.5
5.0
7.5
10.0
12.5
Average degree of HIV infected agents at t=0

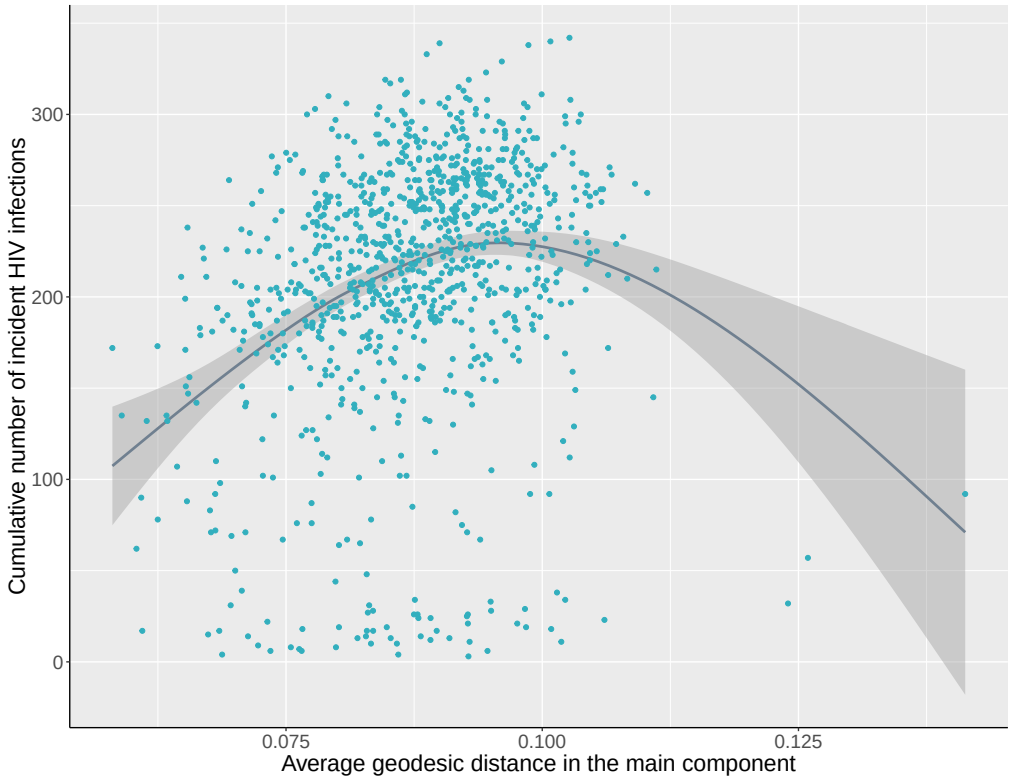

Cumulative number of incident HIV infections
300
200
100
0
0.0
2.5
5.0
7.5
10.0
12.5
Maximum degree of HIV infected agents at t=0

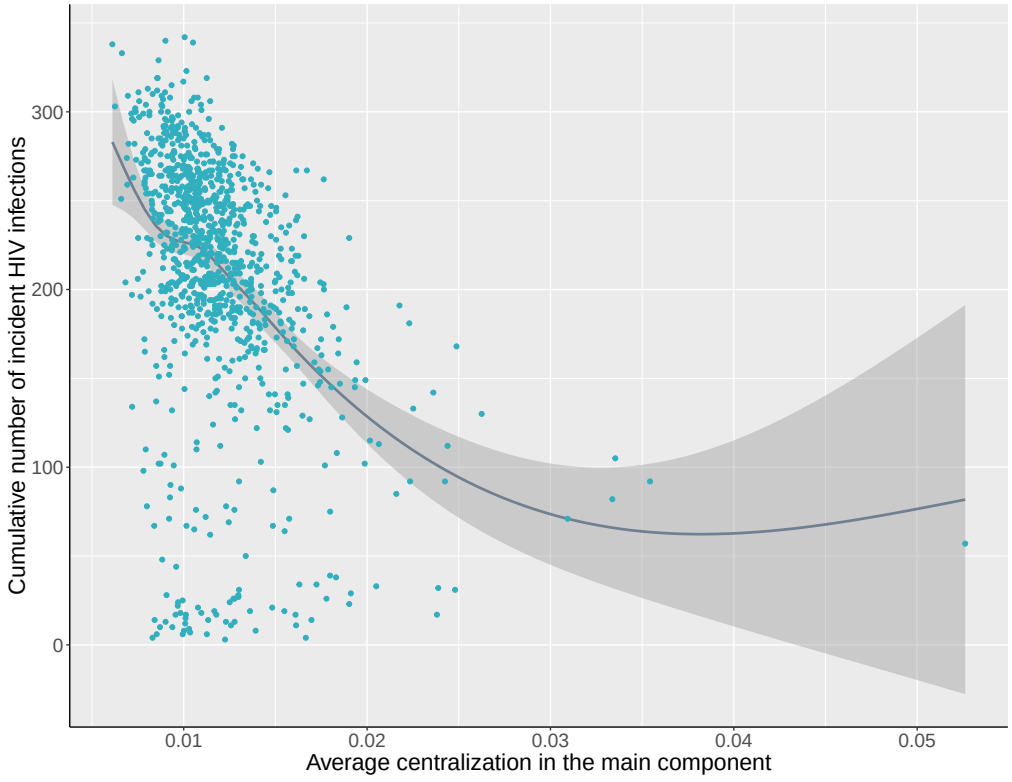

Cumulative number of incident HIV infections
300
200
100
0
0.02
0.04
0.06
Average betweenness centrality in the main component

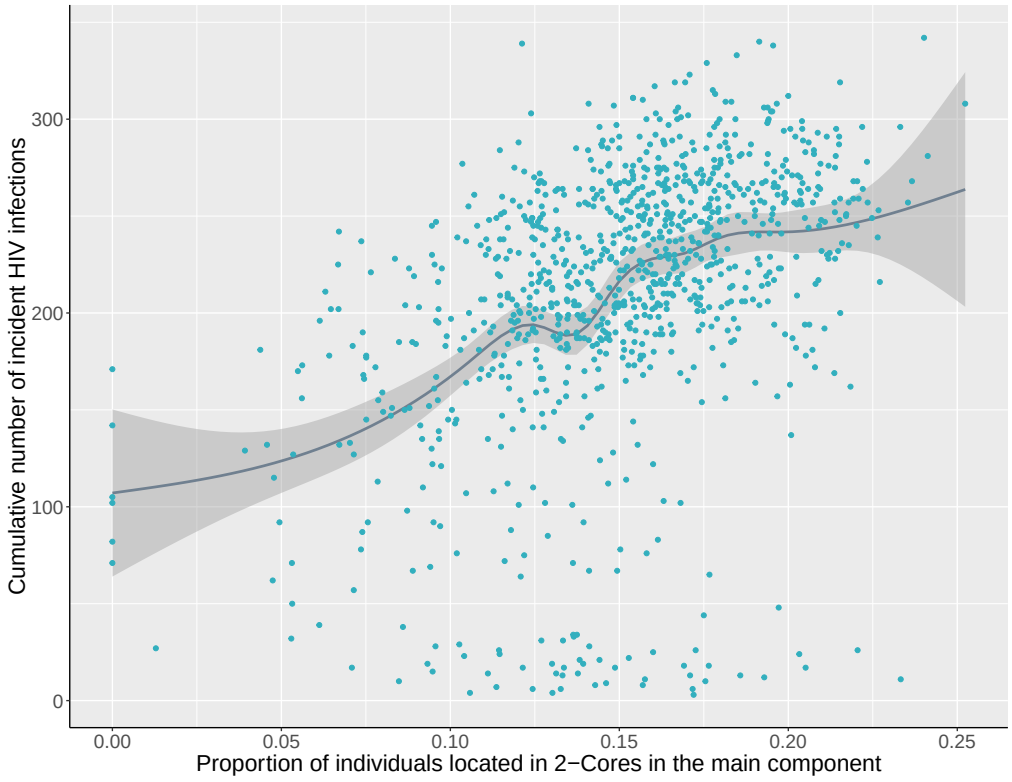

Cumulative number of incident HIV infections
300
200
100
0
0.02
0.04
0.06
Average random walk betweenness centrality in the main component

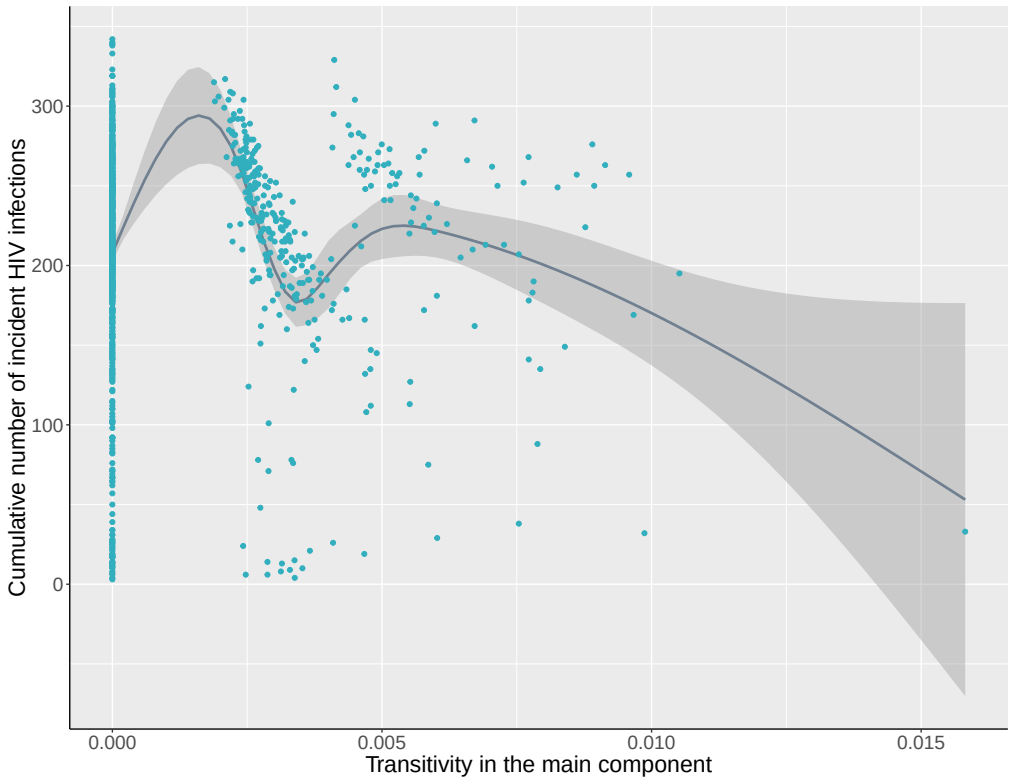

Cumulative number of incident HIV infections
300
200
100
0
200
400
600
800
Number of individuals in the main component


Cumulative number of incident HIV infections
300
200
100
0
0.005
0.010
0.015
0.020
Component density of the main component


Cumulative number of incident HIV infections
300
200
100
0
0.075
0.100
0.125
Average geodesic distance in the main component

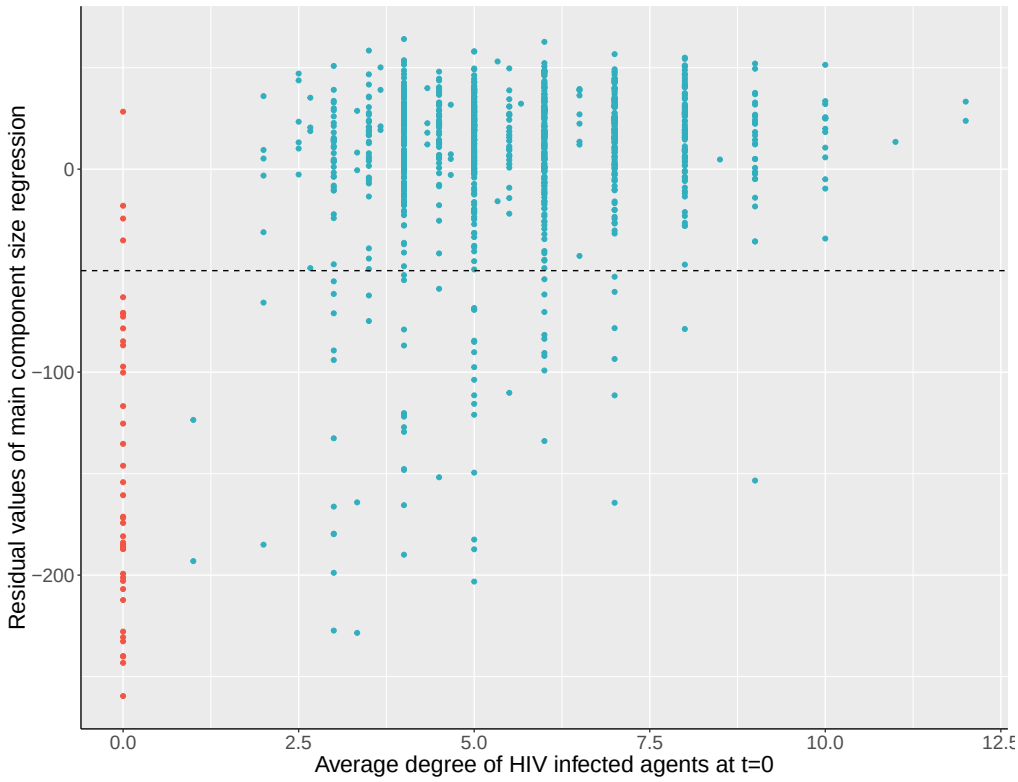

Cumulative number of incident HIV infections
300
200
100
0
0.01
0.02
0.03
0.04
0.05
Average centralization in the main component

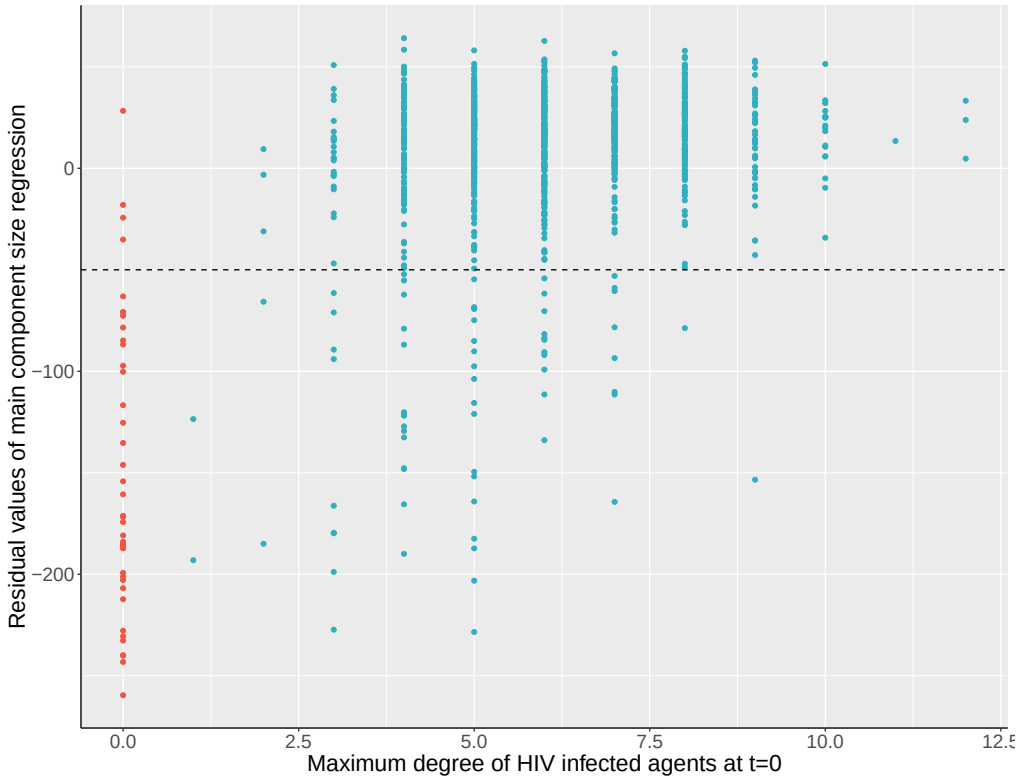

Cumulative number of incident HIV infections
300
200
100
0
0.00
0.05
0.10
0.15
0.20
0.25
Proportion of individuals located in 2−Cores in the main component

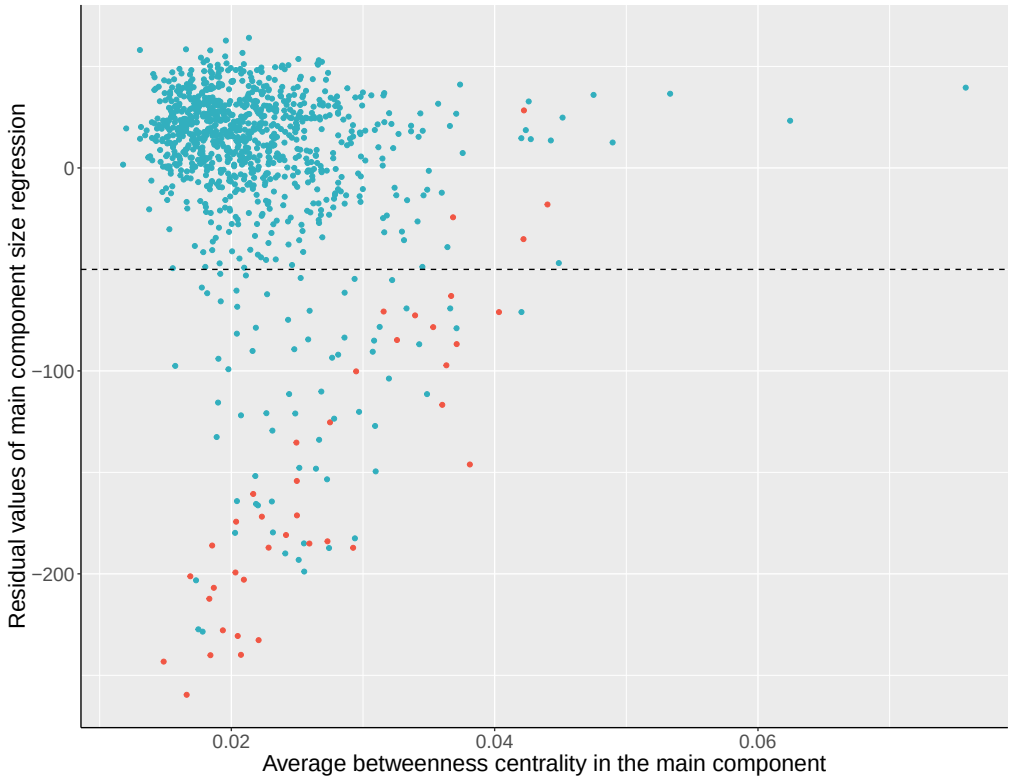

Cumulative number of incident HIV infections
300
200
100
0
0.000
0.005
0.010
0.015
Transitivity in the main component

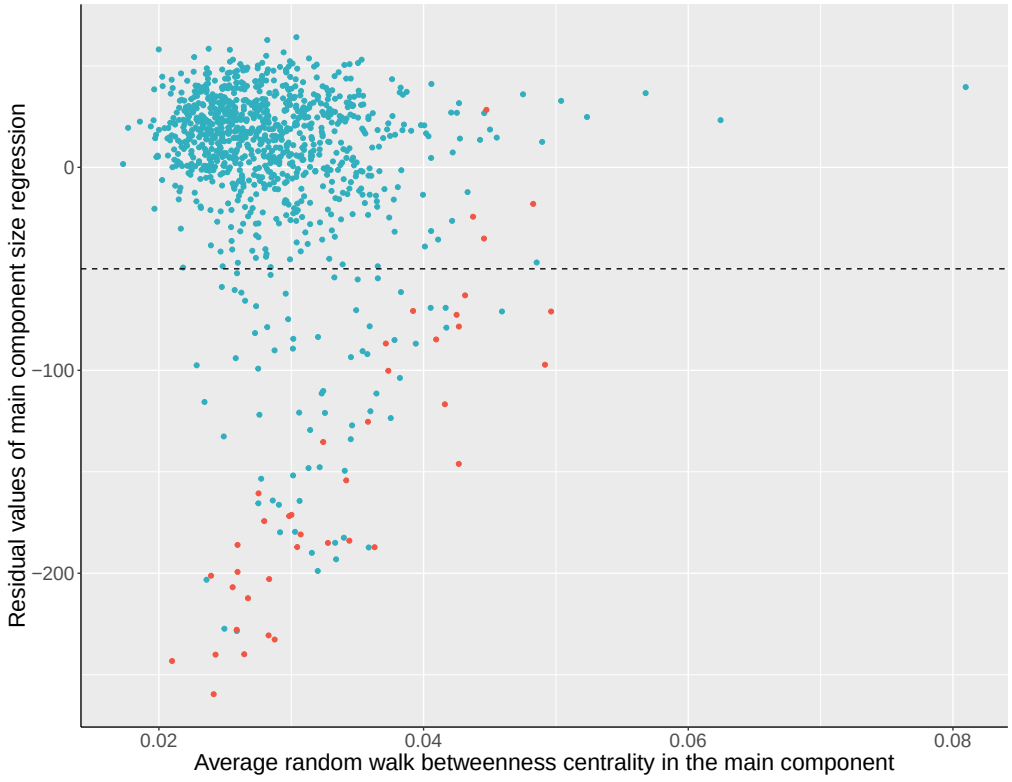
7.3 Secondary Analysis Plots

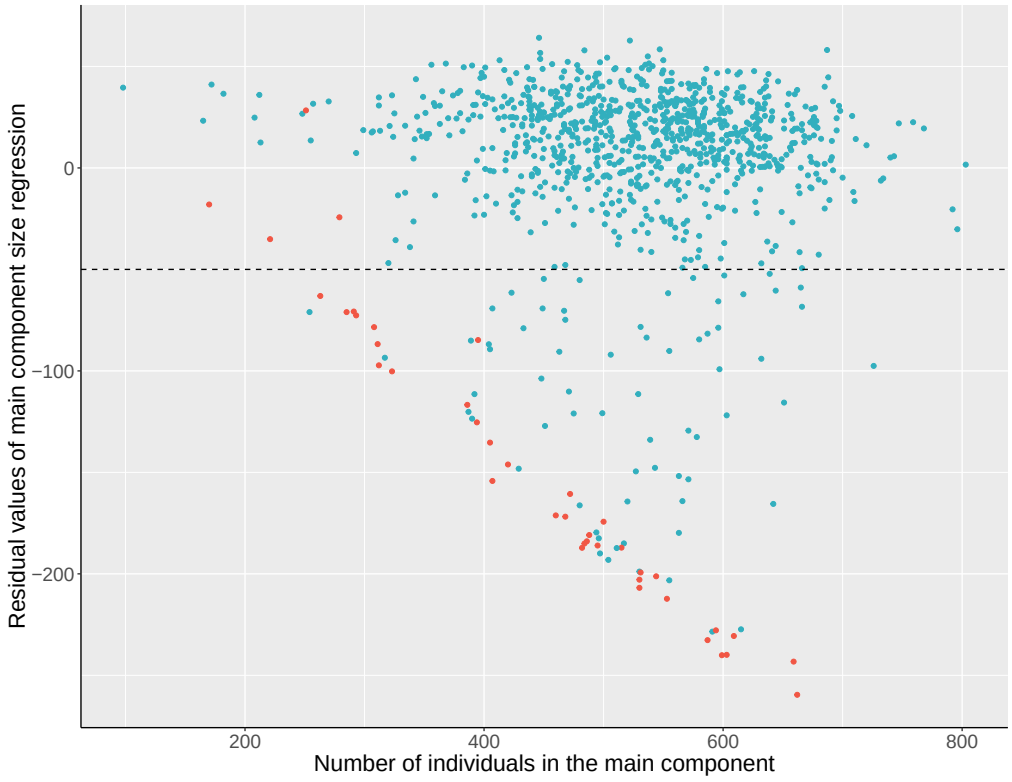
Section 7.3 provides plots of each network metric against the residual values calculated from the component size linear regression model. Each plot includes a dotted line at $y = -50$ as a visual aid in identifying which points classify as having a substantially large residual value. These plots are also colored by whether or not HIV infection is introduced into the main component (red if no, blue if yes).

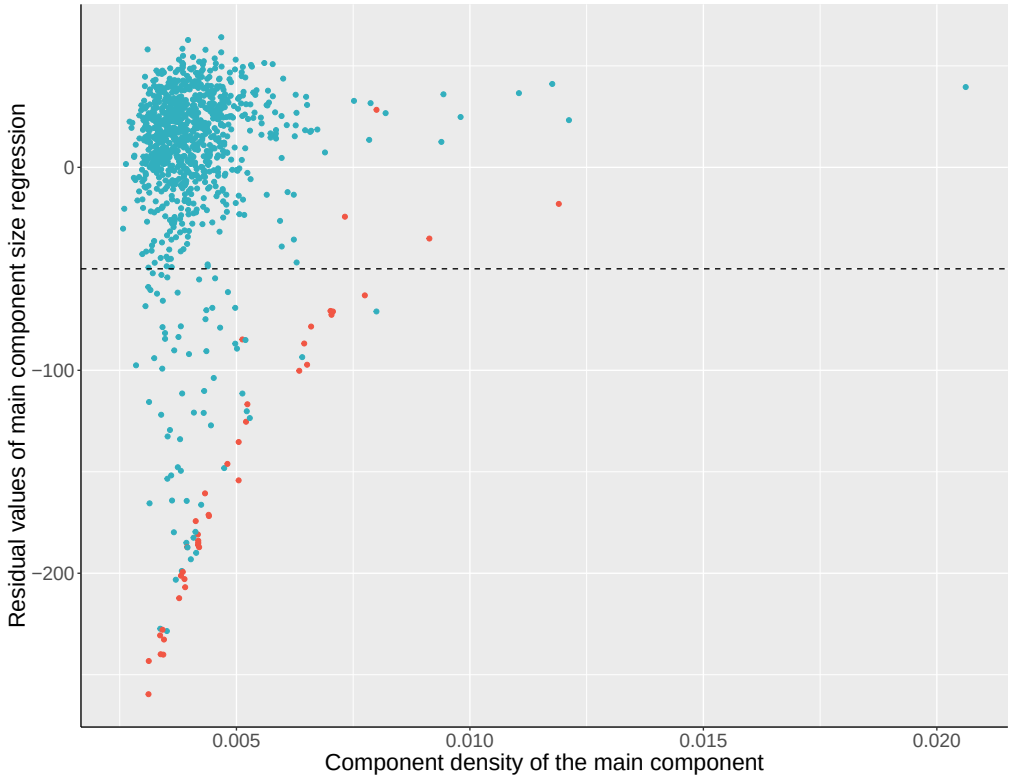

Residual values of main component size regression
0
−100
−200
0.0
2.5
5.0
7.5
10.0
12.5
Average degree of HIV infected agents at t=0

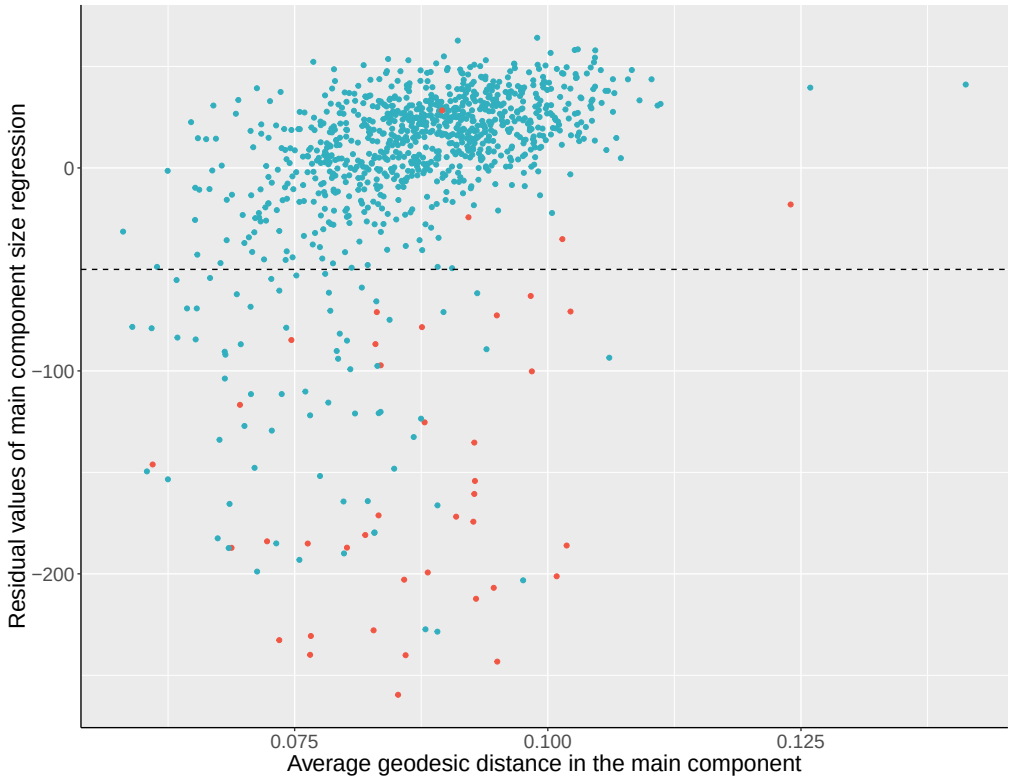

Residual values of main component size regression
0
−100
−200
0.0
2.5
5.0
7.5
10.0
12.5
Maximum degree of HIV infected agents at t=0

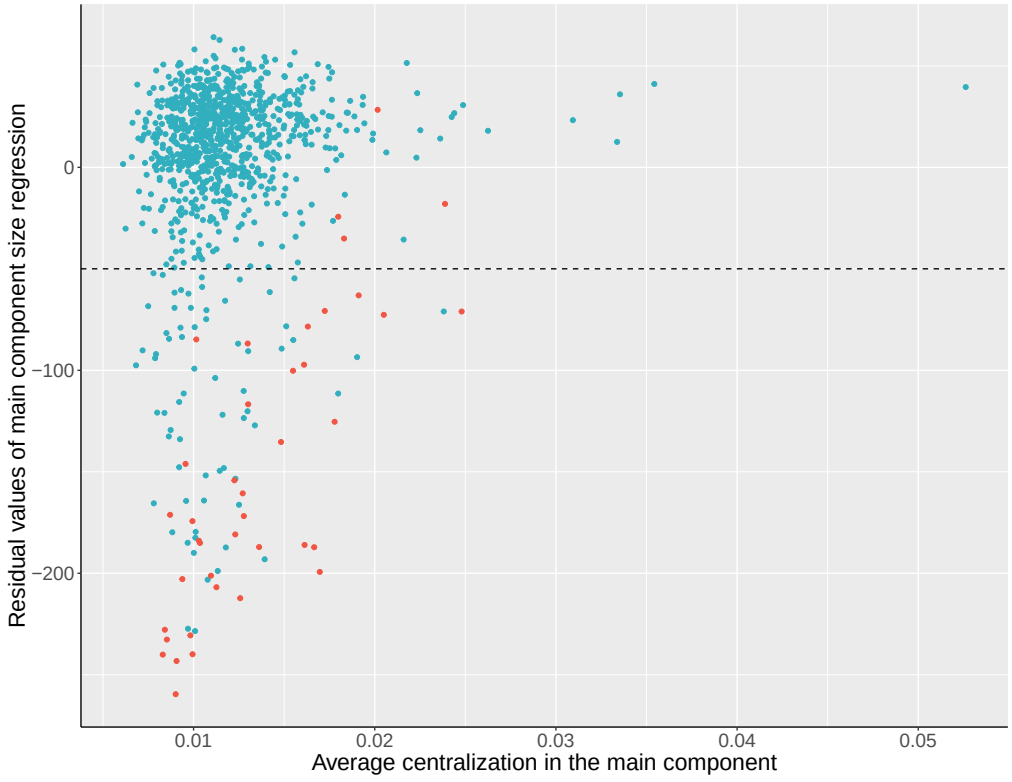

Residual values of main component size regression
0
−100
−200
0.02
0.04
0.06
Average betweenness centrality in the main component

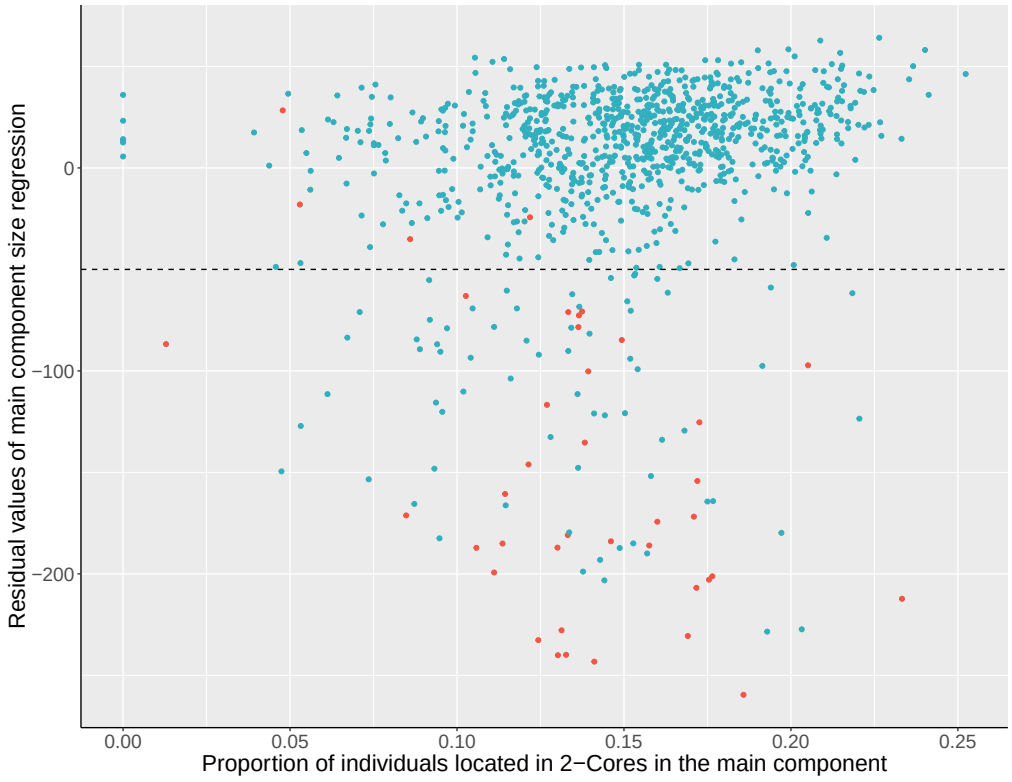

Residual values of main component size regression
0
−100
−200
0.02
0.04
0.06
0.08
Average random walk betweenness centrality in the main component

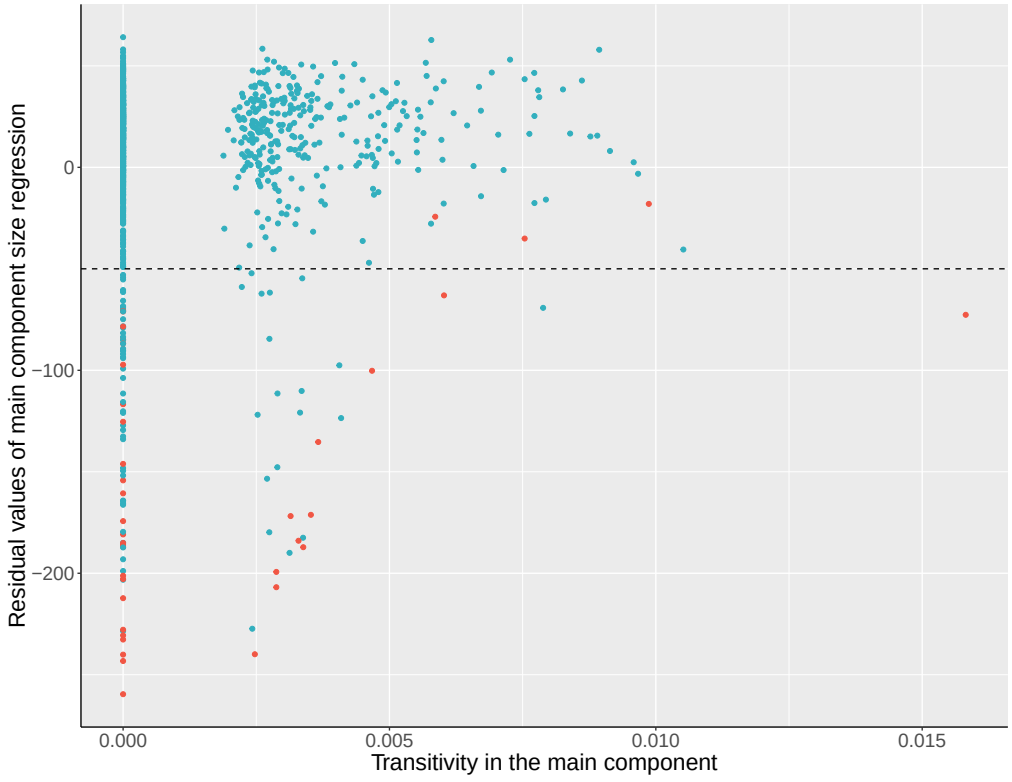

Residual values of main component size regression
0
−100
−200
200
400
600
800
Number of individuals in the main component


Residual values of main component size regression
0
−100
−200
0.005
0.010
0.015
0.020
Component density of the main component


Residual values of main component size regression
0
−100
−200
0.075
0.100
0.125
Average geodesic distance in the main component


Residual values of main component size regression
0
−100
−200
0.01
0.02
0.03
0.04
0.05
Average centralization in the main component


Residual values of main component size regression
0
−100
−200
0.00
0.05
0.10
0.15
0.20
0.25
Proportion of individuals located in 2−Cores in the main component


Residual values of main component size regression
0
−100
−200
0.000
0.005
0.010
0.015
Transitivity in the main component

7.4 Network Analysis Python Code

```
# For networkx package.
import networkx as nx
import statistics

import networkx.readwrite.edgelist

# For network visualization.
import matplotlib.pyplot as pylab

# For network metrics.
import networkx.algorithms.centrality.degree_alg as degree_alg
import networkx.classes.graph as G
import networkx.algorithms.centrality.betweenness as
    betweenness_centrality
import networkx.algorithms.centrality.current_flow_betweenness as
random_walk_betweenness_centrality
import networkx.algorithms.components.connected as
    connected_components
import networkx.classes.function as component_density
import networkx.algorithms.centrality.closeness as
    geodesic_distance
import networkx.algorithms.core as core
import networkx.algorithms.cluster as transitivity

# For command line arguments.
import sys
#print ("This is the name of the script: ", sys.argv[0])
#print ("Number of arguments: ", len(sys.argv))
print (str(sys.argv)) #"The arguments are: "

# Output File Construction
#FILENAME = look up "command line argument"
filename = sys.argv[1] #"edgelist_test_dict.txt"
network_metrics_output_final_ten =
    open("network_metrics_output_final_ten.txt", 'a')

# to type into the VNC :
find *.txt -exec python thesisnetworkanalysis.py {} \;

# ---------------------------------------------------------------
# Build list from edge list output text file.
```

```
def create_edgelist_dict(filename):

#    with open(filename, 'r') as file:
#          rows = ( line.split('\t') for line in file )
#          dict = { row[0]:row[1:] for row in rows }
#     return(listy)

    with open(filename, 'r') as file:
        rows = ( line.split('\n') for line in file )
        final_rows = ( row[0].split() for row in rows)
        listy = [row[0:] for row in final_rows]
    return(listy)

edgelist = create_edgelist_dict(filename)
#print(edgelist)

# ---------------------------------------------------------------------
# Build dictionary from edge list output text file
    with proper indices.

edgelist_dict = dict(enumerate(edgelist))
#print(edgelist_dict)

# ---------------------------------------------------------------------
# Build edge list of all edges to build larger graph.

def create_edgelist(edgelistex):
    ed_list_output = [(v[0], v[7]) for k, v in
        edgelistex.items()]
    return(ed_list_output)

final_edgelist = create_edgelist(edgelist_dict)
#print(final_edgelist)

# ---------------------------------------------------------------------
# Creation of an undirected, simple graph.

G = nx.Graph()
G.add_edges_from(final_edgelist)

#print(G.number_of_nodes())
#print(G.number_of_edges())
```

```
# ------------------------------------------------------------
# Build edge list of just IDU users to build connected component, C.

# first helper
# recall edgelist_dict is our indexed dictionary
# Outputs a list of keys that are relationships that
#    include at least one IDU
def pull_IDU_key_value_pairs(edgelist_dictionary_input):
    listyy = []
    for k,v in edgelist_dictionary_input.items():
        if 'IDU' in v:
            listyy.append(k)
    return(listyy)

IDU_keys_list = pull_IDU_key_value_pairs(edgelist_dict)
#print(IDU_keys_list)


# second helper
# Outputs a subdictionary of all keys that have at least
#   one IDU and all their values
def create_IDU_subdictionary
    (edgelist_dictionary_input, IDU_keys_list_input):
    dict = {k:edgelist_dictionary_input[k] for k in IDU_keys_list_input}
    return(dict)
IDU_subdictionary = create_IDU_subdictionary
    (edgelist_dict, IDU_keys_list)
#print(IDU_subdictionary)


# third helper
# Outputs edgelist from subdictionary in same way
#   as above to create graph I.
def create_IDU_final_edgelist(IDU_edgelistex):
    ed_list_output = [(v[0], v[7]) for k, v in IDU_edgelistex.items()]
    return(ed_list_output)

IDU_final_edgelist = create_IDU_final_edgelist(IDU_subdictionary)
#print(IDU_final_edgelist)

# ------------------------------------------------------------
# Creation of graph of IDU users.

I = nx.Graph()
```

```
I.add_edges_from(IDU_final_edgelist)

#print(I.number_of_nodes())
#print(I.number_of_edges())

# ------------------------------------------------------------------
# Creation of the connected component of interest.

# first helper
# Outputs list of nodes that are in the maximal component of I.
largest_CC_set_of_nodes = max(nx.connected_components(I), key=len)
#print(largest_CC_set_of_nodes)
#print(len(largest_CC_set_of_nodes))

# second helper
# Outputs the list of keys corresponding to all of
#   the nodes identified with largest_CC_set_of_nodes.
#   NOTE : only runs through v[0] because otherwise will double count.
def pull_CC_key_value_pairs(CC_edgelist_dictionary_input,
largest_CC_set_of_nodes_input):
    listyy = []
    for k,v in CC_edgelist_dictionary_input.items():
        for i in largest_CC_set_of_nodes_input:
            if str(i)==v[0]:
                listyy.append(k)
    return(listyy)

CC_keys_list = pull_CC_key_value_pairs
    (edgelist_dict, largest_CC_set_of_nodes)
#print(CC_keys_list)
#print(len(CC_keys_list))

# third helper function
# Outputs a subdictionary of the keys identified above
#   (ie all edges included in connected component) and
#   appending all their values.
def create_CC_subdictionary
    (CC_edgelist_dictionary_input, CC_keys_list_input):
    dict = {k:CC_edgelist_dictionary_input[k] for k in CC_keys_list_input}
    return(dict)
CC_subdictionary = create_CC_subdictionary(edgelist_dict, CC_keys_list)
#print(CC_subdictionary)

# fourth helper
```

```
# Outputs edgelist from subdictionary in same way as above to
#   create graph C.
def create_CC_final_edgelist(CC_edgelistex):
    CC_ed_list_output = [(v[0], v[7]) for k, v in CC_edgelistex.items()]
    return(CC_ed_list_output)

CC_final_edgelist = create_CC_final_edgelist(CC_subdictionary)
#print(CC_final_edgelist)

# ------------------------------------------------------------------
# Creation of graph of largest connected component.

C = nx.Graph()
C.add_edges_from(CC_final_edgelist)
#print(C.nodes())
#print(C.number_of_edges())

# ------------------------------------------------------------------
# ------------------------------------------------------------------


#Creation of an undirected, simple graph.

#G = nx.read_edgelist(filename, nodetype = int)

# Define the initial infection.
#initial_infection = 1703

# Creation of node and edge lists.
#G.add_nodes_from([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
#G.add_edges_from([(1, 2), (1, 3), (1, 4), (1, 5),
#   (1, 6), (1, 7), (1, 8)])
#G.add_edges_from([(11, 12), (2, 3), (8, 9)])

# Define the connected component of interest
#   (i.e. the connected component where the initial
# infection (i) is introduced.

#C = nx.Graph()

# Pull nodes that are connected to i and write into a list,
#   list_connected_nodes.
#list_connected_nodes =
#   list(connected_components.node_connected_component
```

```
#    (G, initial_infection))
#C.add_nodes_from(list_connected_nodes)

# Finish constructing the connected component, C by grabbing
#    all edges each node in C had in G.
#list_connected_nodes1 = list(connected_components.node_connected_component
#    (G, initial_infection))
#for i in list_connected_nodes1:
#     C.add_edges_from(G.edges(i))

# Print edge list for C.
# C.edges()

# ------------------------------------------------------------------

# Tests
#print(G.number_of_nodes())
#print(G.number_of_edges())
#print(C.number_of_nodes())
#print(C.number_of_edges())
#print(G.adj)

# ------------------------------------------------------------------
# ------------------------------------------------------------------

# [1] Degree centrality
# Def : the number of nodes that each node is connected to.
# degree_alg.degree_centrality(G) --> produces fraction of nodes,
#    not integer
#    Outputs dictionary of nodes with degree centrality (fraction)
#         as the value.
#    Num : number of neighbors, Denom : n-1 where n is the number
#         of nodes in the graph.
# We want:
#    - degree of vertex where initial infection is introduced
#    - avg degree of all HIV postive vertices

# First helper
# Outputs a dictionary of nodes with int degree values
def int_degree_dict(dict_input):
    dict = {key:value*(len(G.nodes())-1) for
        key,value in dict_input.items()}
    return(dict)
```

```
final_int_degree_dict = int_degree_dict
    (degree_alg.degree_centrality(G))
#print(final_int_degree_dict)

# Second helper
# Outputs list of nodes that are HIV positive
def pull_HIV_positive_nodes(int_degree_dict_input):
    listyy = []
    for k,v in int_degree_dict_input.items():
        if 'True' in v[6]:
            listyy.append(v[0])
        if 'True' in v[13]:
            listyy.append(v[7])
    return(listyy)

HIV_positive_nodes_list = pull_HIV_positive_nodes(edgelist_dict)
#print(HIV_positive_nodes_list)
#print(len(HIV_positive_nodes_list))

# New helper
# pull only nodes in CCC out of HIV_positive_nodes_list
def pull_HIV_CC_nodes_list(HIV_positive_nodes_list, CC_nodes_list):
    listy = []
    for i in HIV_positive_nodes_list:
        if i in CC_nodes_list:
            listy.append(i)
        else: pass
    return(listy)

HIV_CC_nodes_list = list(set(pull_HIV_CC_nodes_list
(HIV_positive_nodes_list, C.nodes())))
#print(HIV_CC_nodes_list)

# Third helper
# Outputs dictionary that is a subditionary of
#   final_int_degree_dict with only HIV positive nodes
#   and their int degree
def HIV_positive_int_degree_dict
    (int_degree_dict, HIV_positive_nodes_list_input):
    dict = {m:int_degree_dict[m] for m in HIV_positive_nodes_list_input}
    return(dict)

HIV_positive_int_degree_subdictionary = HIV_positive_int_degree_dict
    (final_int_degree_dict, HIV_CC_nodes_list)
```

```
#print(HIV_positive_int_degree_subdictionary)
#print(len(HIV_positive_int_degree_subdictionary))

# Our two desired outputs
# [A] Average degree of all+ HIV positive vertices at t=0.

def avg_degree_HIV_positive_nodes_calc
    (HIV_positive_int_degree_subdictionary):
    if len(HIV_positive_int_degree_subdictionary) > 0:
        return(sum(HIV_positive_int_degree_subdictionary.values())
        len(HIV_positive_int_degree_subdictionary))
    else:
        return(0)


avg_degree_HIV_positive_nodes = avg_degree_HIV_positive_nodes_calc
    (HIV_positive_int_degree_subdictionary)
#print(avg_degree_HIV_positive_nodes)

# [B] Max degree of all of the HIV positive nodes at t=0.

def max_degree_HIV_positive_nodes_calc
    (HIV_positive_int_degree_subdictionary):
    if len(HIV_positive_int_degree_subdictionary) > 0:
        return(max(HIV_positive_int_degree_subdictionary.values()))
    else:
        return(0)

max_degree_HIV_positive_nodes = max_degree_HIV_positive_nodes_calc
    (HIV_positive_int_degree_subdictionary)
#print(max_degree_HIV_positive_nodes)

# [2] Betweenness centrality
# Def : the fraction of shortest paths between some pair of nodes,
#   n1 and n2, that pass through the node of interest.
# betweenness_centrality.betweenness_centrality(C)
#   Outputs dictionary of nodes with betweenness centrality
#   as the value. Only takes in connected graphs.
# We want : average betweenness centrality of the component
#   containing the vertex through which initial infection is
#   introduced

# Outputs a float representing the average betweenness
#   centrality of component containing i.
```

```
average_betweenness_centrality = float(
    sum(
    betweenness_centrality.betweenness_centrality(C).values()))
    / len(
    betweenness_centrality.betweenness_centrality(C))

# [3] Random Walk Centrality
# Note: Referred to as current flow in networkx.
# Def : the number of times a random walk from n1 to n2 passes
#   through the node of interest averaged over all nodes.
# random_walk_betweenness_centrality.
#   current_flow_betweenness_centrality(C)
#   Outputs dictionary of nodes with random walk betweenness
#   centrality as the value. Only takes in connected graphs.
# We want : the average random walk betweenness centrality of
#   component containing i.

# Outputs a float representing the average random walk betweenness
#   centrality of component containing i.

average_random_walk_betweenness_centrality = float(
    sum(
    random_walk_betweenness_centrality.
        current_flow_betweenness_centrality(C).values())) / len(
        random_walk_betweenness_centrality.
        current_flow_betweenness_centrality(C))


# [4] Component Size
# Def : a connected component of a graph as a subgraph of a
#   simple graph G in which every vertex is connected to
#   every other vertex in the subgraph by a path.
# Def : the number of nodes in the connected component that
#   contains i
# Outputs an int.

component_size = C.number_of_nodes()

# [5] Component Density
# Def : the number of edges in the graph divided by the
#   number of total possible edges the graph might have.
# Outputs a float.
```

```
component_density = component_density.density(C)

# [6] Geodesic Distance
# Def : the number of edges on the shortest path between
#   two vertices. We want the average geodesic distance for the
#   component containing i.
# geodesic_distance.closeness_centrality(C)
#   Outputs a dictionary of nodes with closeness_centrality as the
#       value. Reciprocal of how defined in latex document.
# We want : the average geodesic distance in the component where
#   the initial infection is fist introduced

# Outputs a float representing the average geodesic distance in
# the component where the initial infection is fist introduced

average_geodesic_dist = float(
    sum(
    geodesic_distance.closeness_centrality(C).values())) / len(
    geodesic_distance.closeness_centrality(C))


# [7] Centralization
# Idea : "used to calculate the degree to which the network was
#   centralized around one or a few actors," "value reflects the
#   extent to which the network resembles a maximally centralized
#   network in which all network members are connected through one
#   maximally central actor"
# Def : CentD = sum [(largest observed degree index) -
#                    (all degree indices in set)]
#               /  [(len(G.nodes)-1)(len(G.nodes)-2)]

# first helper
def int_degree_dict(dict_input):
    dict = {key:value*(len(C.nodes())-1) for
    key,value in dict_input.items()}
    return(dict)

# definitions
final_int_degree_dict = int_degree_dict(degree_alg.degree_centrality(C))
max_degree_val = max(final_int_degree_dict.values())
degree_vals = final_int_degree_dict.values()

# second helper function, not actually currently in use
# def subtract_dict(dict_input1):
```

```
#     dict = {key:max_degree_val-value for key,value in
#     dict_input1.items()}
#     return(dict)

#subtracted_dict = subtract_dict(final_int_degree_dict)

centralization = ((max_degree_val * len(C.nodes())) - sum(degree_vals))
                 / ((len(C.nodes())-1)*(len(C.nodes())-2))


# [8] k-cores
# Def : a maximal subgraph of G that contains vertices with
#   degree at least k core.core_number(C)
#   Outputs dictionary of nodes with the core number for each vertex.

# Outputs a float representing the proportion of nodes in 2-cores
#   in component containing initial infection.

number_two_cores = ((sum(1 for x in core.core_number(C).values()
                    if x==2)))
#print(number_two_cores)
#print(len(C.nodes()))

average_proportion_2_cores = float(number_two_cores) / len(C.nodes())
#print(average_proportion_2_cores)



# [9] Transitivity
# Def : the proportion of all triads that exhibit closure in
#   the network
#   Outputs a float representing 3 (triangles / triads).

transitivity = transitivity.transitivity(C)

# ----------------------------------------------------------------

# How to display the network. pylab.show() is necessary if
#   not in the interactive mode. Enter into interactive mode by
#   typing ipython -pylab in the cmd.

options = {
    'node_color': 'grey',
    'node_size': 1000,
```

```
    'width': 3,
}
#nx.draw_shell(C, with_labels=True, **options)
#pylab.show()

# ---------------------------------------------------------------------


# Writes to the file path.png in the local directory.
#plt.savefig("path.png")


# ---------------------------------------------------------------------

# Output file

network_metrics_output_final_ten.write("{filename}\t{adHIVpn}
    \t{mdHIVpn}\t{abc}\t{armbc}\t{cs}\t{cd}\t{agd}\t{cent}\t{ap2c}
    \t{tran}\n".format(
    filename=filename,
    adHIVpn=avg_degree_HIV_positive_nodes,
    mdHIVpn=max_degree_HIV_positive_nodes,
    abc=average_betweenness_centrality,
    armbc=average_random_walk_betweenness_centrality,
    cs=component_size,
    cd=component_density,
    agd=average_geodesic_dist,
    cent=centralization,
    ap2c=average_proportion_2_cores,
    tran=transitivity))
```