## Representing Unlabelled Networks

The aspect that makes studying these networks difficult and so also interesting is that the networks are made of arbitrarily labelled nodes and so are effectively unlabelled. Generally, comparing labelled networks is straightforward and can be done by comparing adjacency matrices. Unlabelled networks cannot easily be compared with adjacency matrices because arbitrary labelling means which nodes to compare is not clear. Similarly relabelling any irregular network will result in a non-zero difference according to the adjacency matrix between the original and relabelled networks despite no structural changes in the connectivity. This section details some possible approaches to quantify the difference between unlabelled networks to better understand how to search the space and how point packing can be used to intelligently initialize other optimization algorithms.

### Numerically representing graphs

The sole requirement for a point packing is a minimum distance. In the simplest cases this could be a Hamming distance (error correcting codes) or a Euclidian distance, but any distance metric can be used. While any valid distance metric can be used, there are important implications about the representation of the problem in more complex scenarios. While the distance between two points in Euclidian space is straight forward, the distance between networks is not so simple, especially if the networks are arbitrarily labelled. If graphs are represented in a way that permits some calculatable distance between graphs, diverse collections of graphs can be packed. While simple by design, choosing a representation is complex and has large impact on outcome. There is no singular correct representation and many different good options that must be evaluated. In this section, some possible representations will be examined, and their strengths and weaknesses highlighted. It is important to note that representations are examined through the lens of the problem at hand and may benefit from revaluation for other problems. All representations are evaluated here on undirected, unlabelled, unweighted, connected, static graphs. Because graphs are unlabelled, nodes are ordered in many of the representations to avoid comparing differently labelled isomorphisms. The representations mentioned here do have one drawback in that it is possible to measure a distance of zero between different graphs. These cases present rarely and are often not issues from the lens of point packings as graphs that show this phenomenon are very similar to each other. In addition, graphs that have a non-zero distance cannot be identical.

### Diffusion Characters

Graphs can be compared to each other using diffusion characters. Diffusion characters create a square matrix of each node’s pairwise connectivity by adding unique gas at each node and simulating diffusion. Half of the gas is removed from each node at each time step until the system reaches equilibrium. The square matrix is collapsed into a vector by calculating column wise entropy of gases accumulating at each node. It is possible to point pack diffusion character space (Rn) using Euclidian distance on the entropy vector. Diffusion characters are expensive to compute as they require simulation to compute. The process can be modified to improve runtime by only deploying a unique gas at a subset of the nodes. This creates a partial diffusion character, trading speed for the quality of the reading.

### Page Rank

Graphs can be compared to each other using Google’s PageRank algorithm. PageRank is used to rank webpages that are related to a given query. Webpages related to the query are used to assemble a network based on their linking structure (literal hyperlinks between pages). Pages are given a rank based on a value that describes the traffic received relative to the rest of the network. This differs slightly from diffusion characters in that it only measures volume of traffic through a node, not specifically where that traffic comes from. As a result, PageRank cannot distinguish between regular graphs of different degree as all nodes have the same traffic. Traffic at each node is measured relative to every other node in the network so PageRank lies on the hyperplane of constant sum one. PageRank generates a vector of values in (Rn) and distance between them can be found with Euclidean Distance. PageRank is moderately expensive to compute, but it is well implemented and accessible in many popular libraries.

### Degree Sequence

The degree sequence of a graph is the ordered sequence of the number of neighbors at each node (Zn) sorted in descending order. This representation facilitates distance calculation using Hamming or Euclidean distance. The simplicity of this representation means that it is computationally very fast and inexpensive, but it may suffer from being an oversimplification as it only accounts for direct neighbors. There exist non-isomorphic graphs that have the same degree sequence and so are undistinguishable from each other through the lens of this representation.

### Summary Statistics of Degree Sequence

Degree sequence gives rise to metrics based on summary statistics of degree sequence. These scale better than even the degree sequence because they do not require sorting of the sequence. In addition, while degree sequence length grows as the number of nodes in the graph increases, the summary statistics do not. They can easily be paired together to create 2 and 3 dimensional spaces that are easy to visualize. They do however suffer even more from the same oversimplification issue as different degree sequences can have the same summary statistics, meaning it may fail to identify differences between certain graph structures. This relationship is particularly interesting as mean, standard deviation and skew are the first, second and third moment (respectively) of the degree sequence. An important question to address is if there are any relationships between these representations and if they change depending on the size of the network.

### Comparing Representations

In the process of determining a suitable representation for comparing graphs, many possibilities presented themselves. When similarities between the representations were discovered, it became crucial to examine and understand these similarities. One question that presented itself was if equivalencies between representations existed for all graph sizes. This experiment was created as a due diligence to establish if these relationships would hold for all graph sizes. The secondary benefit is that it also helps to quantify the strength of these relationships. The representations explored here include diffusion characters, page rank and degree sequence. In addition to these some summary statistics of the degree sequence are also studied, including mean, standard deviation, and skew. Pairwise distances between the graphs are calculated using Euclidian distance.

#### Graph Representation Relationship Experiment

Having many possibilities for representing graphs motivates an investigation into the relationship between these different representations. This section documents the creation and documentation of an experiment to investigate the relationships between the representations. One initial issue is selecting a robust way to assess these relationships. The reason this is difficult is because there is a possibility of relationships between representations changing as the size of the graphs increases. Connections between representations present on small graphs may not persist when graphs are large and vice versa. To combat this an experiment to test the relationships between representations at various graphs sizes was designed. Fifty random power law cluster graphs are generated for a variety of graph sizes ranging from 5 to 325 nodes. Pairwise distances between graphs of the same size are calculated for each of the different representations. A correlation matrix is then generated of the pairwise distances of each representation, this is performed for each graph size. If representations largely agree with each other (i.e. graphs that are far apart in one representation are also far apart in the other representation), their correlation will be large. If representations have some disagreement (i.e. graphs that are far apart in one representation are close together in another representation), the correlation will be small. Linear regression is then performed for each representation comparison where the independent variable is the number of nodes, and the dependent variable is the correlation between the pairwise distances of each representation.

The result is one simple linear regression line for each representation comparison. This provides an estimation of the baseline correlation (β0) and the change in correlation as graph size increases (β1). Confidence intervals can be used to determine if graph size impacts the relationship between representations as well as the degree to which the representations agree with each other. Typically, when linear regression is used, slopes of zero indicate no relationship. In this case the lack of relationship means that there should be no issue using surrogate representations on larger graphs. The added benefit is that if the data can be modelled by a horizontal line, the y-intercept can be used to roughly estimate the degree to which two representations agree with one another.



Table 1: 95% confidence intervals for the y-intercept (β0) in the linear regression exploring the relationship between graph size in nodes and correlation of pairwise distances of pairs of representations.



Table 2: 95% confidence intervals for the slope (β1) in the linear regression exploring the relationship between graph size in nodes and correlation of pairwise distances of pairs of representations. Cells with values of zero indicate that the interval contains zero, indicating no change in the relationship as the graph increases in size.

The confidence intervals for β0 give an estimate of the correlation between the two representations. It provides a measure for how similar the spaces are after projecting the graphs using the two representations. Some representations are much more highly correlated with each other. Interestingly, both more expensive representations (diffusion characters and PageRank) have simple representations that can capture very similar distances between graphs. This solidifies the possibility of using degree sequence summary statistic space in place of diffusion characters in most cases. If diffusion characters are desperately needed in some scenario, better computational time could be achieved by first optimizing with summary statistic space and only lastly corrected with diffusion characters. This would allow most of the heavy lifting to be performed by computationally inexpensive representations and refined by expensive ones. The estimates for β0 confirm two major insights related to the relationship between expensive (diffusion characters and PageRank) and inexpensive (summary statistic) representations.

#### Diffusion Characters

These estimates indicate that diffusion characters are significantly less correlated to standard deviation of degree sequence than to all other metrics tested. Of the metrics that are more highly correlated with diffusion characters, there is some variability between the estimates, but no statistically significant difference. Although there is no significant difference between these levels, the highest estimate for correlation to diffusion characters is the combination of mean and standard deviation of degree sequence. While this is not significantly different from degree sequence, it provides better scaling when calculating pairwise distances as the number of dimensions is constant and so does not grow as the size of the graph grows. Additionally, two dimensions allows for intuitive visualization.

#### PageRank

The estimate for correlation to the PageRank metric indicate that PageRank is significantly more correlated with degree sequence and skew of degree sequence than it is to the other metrics tested. While skew is not significantly more correlated than degree sequence, the estimate is larger, and it benefits from the same scaling benefits listed in the diffusion character section.

The intervals for β1 indicate that for the most part these relationships between representations do not change as the graph size increases. Confidence intervals that contain zero simply show zero as the estimated slope because it cannot be proven otherwise. Some intervals for the slope do not contain zero. These estimates are values with very small magnitudes and slightly negative. This indicates that the relationship may decay in graphs larger than ~10,000 nodes; however, generally extrapolating that far from the sample is not advised. Small negative estimates for the slope indicate that these relationships should be reassessed if significantly larger graphs are needed but are indistinguishable from zero on the sizes of graphs sampled here.

### Surrogate Representations

The primary reason for exploring this space is that the computational complexity of simulating an epidemic in a defined network is high. This means that optimizing for networks that facilitate types of epidemic spread is expensive because fitness updates are computationally intensive. These surrogate representations can be used to help navigate network space by suggesting graphs that are different or similar to candidate networks depending on whether exploration or exploitation is required. These tactics can offload some of the complexity of optimising networks for epidemics simulation, reducing runtime and making this process significantly more feasible. Diffusion characters are expensive to calculate, but this analysis shows that a significant portion of the information that can be used to differentiate graphs with diffusion characters is preserved in some simpler metrics. While this is true, diffusion characters still offer higher quality representation. To circumvent the cost of computing diffusion characters, networks can be point packed using cheaper metrics like mean and standard deviation of degree sequence with a final pass performed using diffusion characters using mean and standard deviation of degree sequence as a surrogate representation for the majority of the evaluations. Of course, the degree to which each representation is used is easily tunable. This allows for the best of both worlds, improved speed and confirmation of high-quality solutions.

## Enforcing Exploratory Behaviour in Evolution with Point Packings

The motivation for exploring unlabeled network search space was to investigate the underlying reason that the network optimizer is unable to find networks that are substantially different from the network it is initialized with. Analysis of the search space shows that the fitness landscape is incredible rugose. This rugosity makes it difficult to traverse through the editing commands causing exploitation to take priority over exploration, shown in Figure 1. The result is a preference for finding local optima rather than a global optimum. This behaviour is particularly problematic because of the cost of explicit simulation. Because the fitness function is expensive to compute increasing the tolerability of intermittently finding lower quality solutions to permit the discovery of new maximum fitness values become computationally prohibitive. The mix of representations presented enables navigation of the search space without epidemic simulation. These representations can be used to suggest networks that are substantially different from those already simulated, forcing the evolutionary algorithm to venture into previously unexplored areas of the search space. While this method does not guarantee the global optimum will be found, it gives higher credibility that the optimum found is a global optimum because more of the space has been searched.

Chart

Description automatically generated

Figure 1: Exploration issues in evolutionary optimization of network structure to mimic daily case numbers. Two variants of initial graphs were used: power-law cluster (yellow) and ring graphs (black). Initial graphs are shown as exes and the evolved graphs are shown as circles. These graphs are plotted on the first two principal components of their diffusion characters. Evolved graphs show little propensity to move far from their initialization. This is in line with the informal assessment that at no point in time during the optimization were drastically different solutions found.

Point packings are well spaced-out sets of points and have a diverse set of uses. Here, they are used to force a diverse set of networks to use for initialization, but other uses include error correction and fast clustering. Network point packings are created by adding networks to the packing if they are at least some minimum distance from all other packing members. Any of the representations mentioned in the previous section are suitable but exploiting the similarity between diffusion characters (highest quality) and summary statistic of degree sequence (computationally easiest) has proved to be very fruitful. The more distance calculations that can be performed with the given computing resources, the better and more complete the coverage of the packing will be. Complete packings enable the best initialization efficiency, maximizing the coverage of a fixed number of initializations. Point packings also require some form of random network generation to take as input. There is great flexibility in the way that networks can be generated. Generating networks throughout the search space as close to uniformly at random as possible is ideal, but not required. It is also not entirely straightforward how to accomplish such a feat and likely to be dependent on the representation that is chosen. Methods that are uniform for one representation may not be uniform in a different representation. Generating graphs for point packing can be done in many ways, but the best results were found using bit sprayers to randomly fill in adjacency matrices. Bit sprayers proved to have a significantly better coverage of the space when compared to graph generators for specific graph models such as Erdős–Rényi, Watts-Strogatz and Power-Law Cluster as is shown in Figure 3.

Point packing can be performed using any graph generator, allowing for modification of the behaviour based on the type of input used. If for example, the desired graphs should be based on an existing graph model, then there is justification to use that model to generate graphs for input. However, you can also test the hypothesis that graphs of a specific model will be superior based on the probability that the optimized bit sprayer graphs could be generated under the given model. Bit sprayers have some tunable parameters that impact the bias of the resultant graph generation, but generally they generate a very diverse collection of graphs. This can be seen in Figure 2 along with a sample packing of graphs. The minimum distance of the packing is used to dictate the size of the packing. A smaller minimum distance allows for more graphs to be fit into the packing. The point packing works like a filter and helps to select graphs that are sufficiently different for each other to avoid redundant optimizations. The minimum distance should be tuned such that the packing roughly contains the maximum number of initializations that are feasible.

Chart, scatter chart

Description automatically generated

Figure 2: Bit sprayer graphs and a sample point packing. Both sets shown are displayed using summary statistic of degree sequence. 10,000 graphs generated randomly using a bit sprayer (left) and a sample point packing from those graphs (right). The packing uses a minimum distance of 13.5, which results in 36 candidate graphs to be used for initialization of the network structure optimizer.

Scatter chart

Description automatically generated with medium confidence

Figure 3: Distribution of random graph generators using summary statistic of degree sequence as representation. Graphs generated using Bit Sprayers (Black) cover the available space much more evenly than the Erdős–Rényi graphs at variety of different parameters (various colours). Erdős–Rényi graph parameters can be modified to achieve a variety of different mean degree but tend to cluster toward central values for standard deviation of degree.

If there is good information that a particular graph model is preferred, the appropriate generator can be used to narrow the scope of possible initializations. In the context of epidemic simulation, power-law cluster graphs are thought to model the connectivity found in human communities and social groups well. Figure 4 shows the impact of initializing evolution with a point packing created using power-law cluster graphs as input. Point packed graphs result in a much greater coverage of the available area, enabling more complete search of the space. Again, it is important to note that this does not guarantee that it will outperform standard initialization on a single run, but that it will outperform it on average. Here, we can see surrogate representations in action because this point packing was performed using summary statistic of degree sequence to compute distance between networks. The separation is still present when the results are visualized with the more comprehensive and expensive diffusion characters.

Chart, scatter chart

Description automatically generated

Figure : Highest fitness networks after optimization for a disease progression profile. Networks are represented by the first two principal components of their diffusion characters. Power-law cluster (yellow), Erdős–Rényi (red) and Watts-Strogatz (blue) graphs are shown as well as a fourth initialization using a point packing with power-law cluster graphs as input (green). The point packed graphs span a significantly larger range across the space than is covered by the same number of initializations with random power-law cluster graphs. Both graphs originating from power-law cluster generation span a significantly larger area of the space than either of the other two generation methods. Interestingly all three models do occupy a unique area of the space.

## Selecting Patient Zero and Initiating the Epidemic

Two other components of simulating epidemic spread through a network present additional issues for accurate simulation. Both causes of these issues are centered around how the epidemic simulation begins. This section will focus on two important decisions: choosing the patient zero and when to start comparing new infections at each time step to the profile of interest. Choosing the patient zero has a different effect depending on the network, but the effect can be quite substantial. Additionally, patient zero arguably becomes a node worth labelling, which introduces the issues of considering a partially labelled network. Other problems arise from the interaction between the chosen disease model and the disease profile the network will be optimized to. Disease models with many states may contain intermediate states prior to infection to model incubation periods for a more realistic simulation. While including these features may improve the quality of the results, they can prolong the period before infection can begin to ramp appropriately. This shifts the simulated daily infection counts, prohibiting high quality matches to aggressive disease profiles. Both the issues of selecting the patient zero and accommodating models with more states will be covered here.

### Selecting Patient Zero

Because of the stochastic nature of simulation, many replicates are required to avoid results that cannot be reproduced. The unlabeled nature of the networks introduces an additional caveat of determining where to start the epidemic. While repeated sampling can be used to get a good idea of how the epidemic will progress on average, the same approach cannot be taken to a patient zero. Randomly selecting the patient zero causes problems for the optimization of the network by convoluting the conditions under which optimization takes place. It is difficult to make alterations to the network that will constitute positive change under all possible patient zeros. Effectively, this will send mixed messages to the fitness function. Once optimization begins, the node used for patient zero should not be changed. In addition, it is not necessary randomize that part of the process.

The node labelled ‘0’ for encoding purposes is no more suitable to be patient zero than any of the other nodes. There is both theoretical and experimental justification that all possible patient zeros must be considered. This is especially true in conjunction with a point packed initialization because the structure of the network is unique regardless of a different labelling. The disease progression starting from a node is not comparable to that found in other networks, otherwise the structure would be similar which point packing prohibits. It is possible for the choice of patient zero within a network not to have any impact on the epidemic. If the network is a regular graph, all nodes can have the same connectivity to the rest of the network. The simplest example of this is a ring graph, functionally all nodes are identical starting positions and any difference between the nodes will disappear with sufficient sampling.

To determine the extent to which the selection of patient zero impacts the spread of the epidemic through the network, an experiment was conducted comparing how the epidemic spread considering each possible patient zero in turn. Because the type of graph is expected to impact the degree to which it is important, this experiment is performed on a collection of point packed graphs to ensure a diverse set of networks is tested. The epidemic simulation is performed 100 times with each node in each network as the patient zero and the RMSE for a profile is recorded. This allows for the study of the variability of the outcomes of simulation across all possible patient zeros. The results for this experiment are shown in Figure 5. While some networks are affected more than others, all networks have the capacity to change the how epidemic simulation proceeds based on which patient zero is chosen. Generally, the networks that have the capacity to match the profile most closely also have large discrepancy of epidemic progression. Patient zero selection has a large impact on the progression of the epidemic.

Chart, box and whisker chart

Description automatically generated

Figure : Investigating the impact of patient zero selection. A point packing of 30 graphs each with 303 vertices are used to ensure a diverse collection of graphs are used. Graphs are point packed using summary statistic of degree sequence space. Boxplots show the distribution of the average RMSE for 100 simulations using each of the 303 vertices as patient zero. RMSE is calculated in this example to the Wellington Dufferin Guelph profile. It is important to note that all these networks are not yet optimized to fit the profile. Assessment of each patient zero will inform which node will remain patient zero during optimization.

Using different nodes as patient zero can have a large effect on how simulation of an epidemic proceeds through a network. The magnitude of this effect is dependent on the network and the desired disease profile. Ultimately, selecting a good patient zero will help orient the downstream optimization to the disease profile. Figure 6 show more detailed information for one of the networks from Figure 5. This shows how not only are some nodes more eligible to be patient zero (for the given profile), but also that some patient zeros are subject to significantly more “randomness” with respect to the outcome of simulation. Ideally, a network will be found such that an epidemic that roughly matches the profile can be reliably simulated. Figure 6 shows an example of a graph that has a few very good candidate patient zero nodes (both in terms of matching and simulation consistency) for the profile in question.

Chart

Description automatically generated

Figure : Side-by-side boxplot of RMSE from profile for 100 samples with each node as patient zero. Nodes are ordered by ascending median value. This is a more detailed view of the information that is used to create the boxplot for Graph 2 in Figure 5. The left most nodes are the best candidates for a patient zero that will produce a disease progression similar to the profile, whereas the right most nodes produce a disease progression that is most different from the profile. For this network and profile combination, there are wide range of outcomes depending on the patient zero that is selected. In this instance, the left most nodes also have the most consistent outcome with the least variability.

### Initiating the Epidemic

Along with selecting the patient zero there are some other issues that must be considered when initiating graph based epidemic simulation for profile matching. Two specific complications are discussed here, but there is some common ground these problems share. Both have to do with avoiding scenarios where exact matching of the profile is impossible. While it is not expected that the profile will be matched exactly, having this not be feasible indicates the possibility for undesirable bias. Two different ways this presented itself was in comparing to profiles later in the progression of the disease and when considering more complex models with additional states. Interestingly, there is one solution that addresses both issues. Both problems become trivial if the profile is matched to the best set of consecutive days, that is the set of days whose infection numbers have the lowest RMSE from the profile of choice.

Initializing the epidemic by setting one individual’s status to infected can pose a problem when matching to profiles that begin with more than one infected individual. This causes some difference from the profile to be inherently present but can also be detrimental by influencing the way the epidemic must ramp up to compensate. The result is unintentionally matching to a different profile than intended. The issue can be circumvented by either changing how the epidemic is initialized or by modifying the profile. Initializing more patient zeros is straightforward, modifying the profile is less so. If a profile represents the daily case numbers for a defined region, then it can feasibly be divided into subregion, the sum of which would be the original profile. Profiles can be split such that all portions of the profile begin with only one infection. Both solutions do introduce additional considerations. Initializing with multiple patient zeros introduce the issue of selecting patient zeros, the number of possibilities for which grow combinatorically. Splitting the profile into separate parts can be done in many ways with no objectively correct solution. The alternative to these approaches is to initialize the epidemic with one patient zero, simulate the epidemic for a fixed number of days longer than the profile and allow profile matching to occur as a sliding window, ultimately selecting the window with the lowest RMSE to the profile. This allows for a burn-in period if required for the epidemic to ramp up at the same rate as the profile.

Accommodating models that are more complex than the SIR model can help simulation more accurately reflect reality. Generally, these more complex models introduce additional states to account for exposure prior to infection as well as asymptomatic individuals. Introducing additional states, particularly between susceptible and infected states can have some unintended consequences for the initialization of simulation. Additional states cause an initial lag in cases. If one individual is infected on the first day (patient zero), the second day will have no new infections because individuals must pass through one or more intermediate states. The consequence is that the profile cannot possibly be matched well during the beginning of the profile and is subject to the same overcompensation discussed earlier. The issue here is that the states are empty when initialized and so are not setup organically. When an individual tests positive and is counted as a daily case, they have likely already infected other individuals (required for anything to spread). Initializing an epidemic and allowing for burn-in to occur ensures that nodes will organically be in various states when comparison to the profile begins.

Figure 7 shows the effect of allowing for a burn-in period to occur. These figures use the same set of networks, profile, and model as Figures 5 and 6. This is a profile that has the first daily case number larger than 1 and the SEE’AIR model. The SEE’AIR model has two states that precede the infected state as well as an asymptomatic state, so it is crucial for the system to organically have nodes assigned to those states at the start of simulation. As is shown, allowing for burn in results overall in a slightly lower average RMSE and is believed to improve the generalizability of the results.

Chart, box and whisker chart

Description automatically generated

Figure : Patient zero experiment with burn-in. This data originates from the same style of experiment that produces figures 5 and 6. The exception is that simulation is performed for longer, enabling a sliding window of comparison to the profile. For any single simulation this necessarily can only reduce the average RMSE, although the randomness of the 100 samples does not guarantee this outcome. This approach also seems to produce fewer outliers.