The probability of edge existence due to node degree: a baseline for network-based predictions

This manuscript (permalink) was automatically generated from greenelab/xswap-manuscript@d0ae072 on August 13, 2019.

Authors

- **Michael Zietz**
  0000-0003-0539-630X · zietzm · ZietzMichael
  Department of Physics & Astronomy, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America; Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America · Funded by ['Roy and Diana Vagelos Scholars Program in the Molecular Life Sciences', 'the Gordon and Betty Moore Foundation (GBMF4552)']

- **Daniel S. Himmelstein**
  0000-0002-3012-7446 · dhimmel · dhimmel
  Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America · Funded by Pfizer Worldwide Research, Development, and Medical; the Gordon and Betty Moore Foundation (GBMF4552)

- **Christopher Williams**
  · chrsunwil
  Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America

- **Michael W. Nagle**
  0000-0002-4677-7582 · naglem · MikeNagle84
  Internal Medicine Research Unit, Pfizer Worldwide Research, Development, and Medical

- **Casey S. Greene**
  0000-0001-8713-9213 · cgreene · greenescientist
  Department of Systems Pharmacology and Translational Therapeutics, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America · Funded by ['Pfizer Worldwide Research, Development, and Medical', 'the Gordon and Betty Moore Foundation (GBMF4552)', 'the National Institutes of Health (R01 HG010067)']

Abstract

Networks of biomedical data rarely consist of all true relationships. Instead, networks contain spurious relationships while omitting actual relationships. How a network deviates from the real set of relationships is often biased according to node degree, resulting from processes such as inspection bias and experimental methods. While degree is subject to potentially substantial biases, link prediction methods can be strongly affected by degree. In the present work, we introduce a network permutation framework to quantify the effect of node degree on network-based methods and prediction tasks. We introduce the “edge prior” to quantify the probability that two nodes are connected based only on their degree. After demonstrating that this prior feature shows excellent discrimination and calibration performance for 20 different biomedical networks (16 bipartite, 3 undirected, 1 directed), we conclude that our prior feature represents a suitable baseline for network link prediction tasks, as performance exceeding the baseline is attributable to factors other than degree alone. Additionally, we propose methods to incorporate network permutation and the edge prior into other predictive methods. Our results highlight the importance of degree for link prediction and provide a way to account for its effects when degree bias may be present. We have released a full implementation of our network permutation method and the edge prior as an open-source Python package on GitHub.

Introduction

Networks contain information about relationships between entities (“edges between nodes”). A node’s degree is the number of relationships it has in the network. Networks contain many nodes, whose degrees can be aggregated to form the network’s degree distribution. Because different nodes can have very different degrees, real networks have highly variable degree distributions (Figure 1).

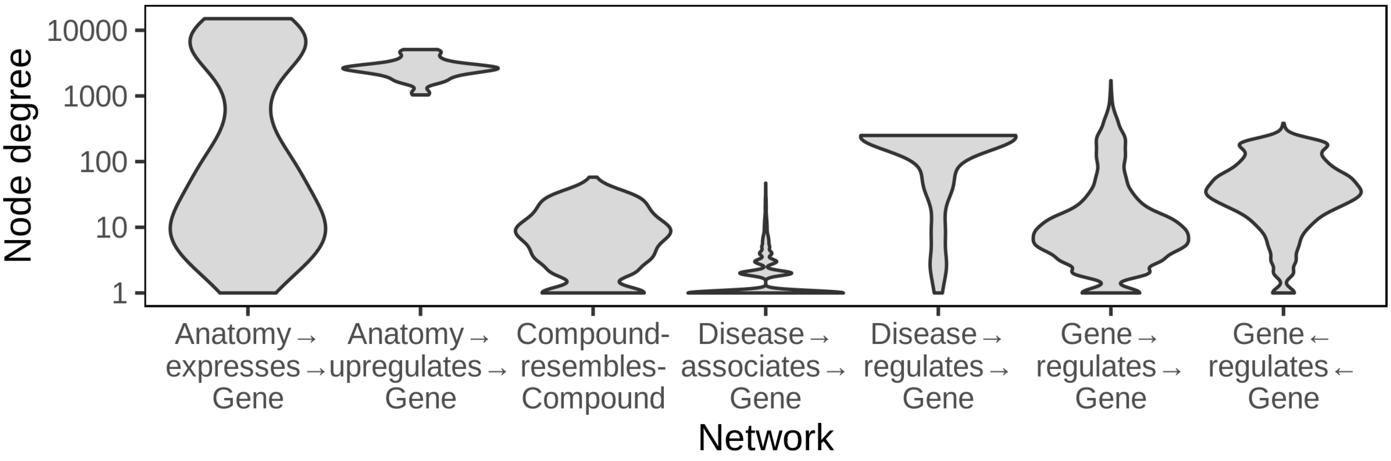


Figure 1: Degree distribution can vary greatly between networks, even between the source and target degree distributions within the same (directed or bipartite) network. Shown are seven degree distributions for six edge type subnetworks of Hetionet [1]. (The final two are the source and target distributions of the gene-regulates-gene directed edge type.)

Degree is an important metric for differentiating between nodes, and it appears in many common edge prediction features [2]. However, reliance on degree can pose problems for edge prediction. Firstly, bias in the data can distort node degree so that degree differences between two nodes may not be meaningful. Secondly, reliance on degree can lead edge prediction methods to make nonspecific or trivial predictions and fail to identify novel or insightful relationships.

Most biomedical data networks are imperfect representations of the true set of relationships. Real networks often mistakenly include edges that do not exist and exclude edges that do exist. How well a network represents the true relationships it attempts to represent depends on a number of factors, especially the methods used to generate the data in the network [3,4,5]. We define "degree bias" as the type of misrepresentation that occurs when the fraction of incorrectly existent/nonexistent relationships depends on the number of connections that nodes make (their "degrees"). Depending on the type of data being represented, degree biases can arise due to experimental methods, inspection bias, or other factors [3].

Inspection bias indicates that entities are not uniformly studied [6], and it is likely to cause degree bias when networks are constructed using hypothesis-driven findings extracted from the literature, as newly-discovered relationships are not randomly sampled from the set of all true relationships. For example, the number of publications mentioning a gene has little correlation with its degree in a systematically-derived protein interaction network [7]. Therefore, the high correlation between number of publications and degree in low-throughput interaction networks is almost entirely the result of inspection bias. This evidence suggests that many poorly studied genes have similar numbers of interactions as the genes scientists have preferentially examined. The consequence of degree bias is that a difference in degree between two nodes may not reflect a difference in the number of true edges. Reliance on degree may be unfavorable depending on the prediction/analysis task being conducted and the magnitude of possible degree bias in the data.

Another reason why a method's reliance on degree can be unfavorable is that degree imbalance can lead to prediction nonspecificity. Nonspecific predictions are made on the basis of generic characteristics rather than the specific connectivity information contained in a network. For example, Gillis et al. [8] examined the concept of prediction specificity in the context of gene function prediction and found that many predictions appear to rely primarily on multifunctionality and could be "potentially misleading with respect to causality." Real networks have a variety of degree distributions (Figure 1), and they commonly exhibit degree imbalance [10,11,9]. Degree imbalance leads high-degree nodes to dominate in the predictions made by degree-associated methods [12], which are effective predictors of connections in some biological networks [13].

Consequently, degree-based predictions are more likely nonspecific, meaning the same set of predictions performs well for different tasks. However, depending on the prediction task, edge predictions between very high degree nodes may be undesired, uninsightful, or nonspecific. Model evaluation is challenging in this context, as nonspecific or trivial predictions need not be incorrect, even if they are not the desired outputs of the predictive model. For example, predicting that the highest degree node in a network shares edges with the remaining nodes to which it is not connected will often lead to many correct predictions, despite the predictions being generic to all nodes in the network. Degree-based features should often be included in the interpretation of predictions to disentangle desired from non-desired effects and to effectively evaluate and compare predictive models.

Degree is important in edge prediction, but it can cause undesired effects. We sought to understand the effect of node degree on edge prediction methods. We introduce a permutation-based framework to find edge existence probabilities due to node degree and to quantify the contribution of degree to edge prediction methods. This method allows edge predictions to be evaluated in the context of degree and its effects on the prediction task. Our results demonstrate that degree-associated methods are very effective for reconstructing a network using a subsampled holdout but ineffective for predicting edges between distinct degree distributions. Using a number of different networks, we provide evidence that degree has a strong effect on the probability of edge existence and that our "edge prior" feature best quantifies this probability.

Methods

Network permutation

Network permutation is a way to produce new networks by randomizing the connections of an existing network. Specialized permutation strategies can be devised that randomize some aspects of networks while retaining other features. Comparing between permuted and unpermuted networks gives insight to the effects of the retained network features. For example, an edge prediction method that has superior reconstruction performance on a network compared to its permutations likely relies on information that is eliminated by permutation. Conversely, identical predictive performance on true and permuted networks indicates that a method relies on information that is preserved during permutation. Network permutation is a flexible framework for analyzing other methods, because it generates networks with identical formats to the original network. We propose using network permutation to isolate degree and determine its effects in different contexts. Degree-preserving network permutation obscures true connections and higher-order connectivity information while retaining node degree, and thereby, the network's degree sequence. Thanks to the flexibility of permutation, our framework can quantify the effect of degree on any network edge prediction method.

XSwap algorithm

Hanhijärvi, et al. presented XSwap [14], an algorithm for the randomization ("permutation") of unweighted networks (Figure 2A). The algorithm picks two existing edges at random, and if the edges constitute a valid swap, exchanges the targets between the edges (Figure 1). This process is repeated many times until the maximum number of steps has been reached. In general, the maximum number of steps should be chosen to be sufficiently large that the fraction of original edges retained in the permuted network is near its asymptotic value for a large number of steps.

To allow greater flexibility, we modified the algorithm by adding two parameters, "`allow_loops`", and "`allow_antiparallel`" that allow a greater variety of network types to be permuted (Figure 2B). Specifically, two chosen edges constitute a valid swap if they preserve degree for all four involved nodes and do not violate the above condition options. The motivation for these generalizations is to make the permutation method applicable both to directed and undirected graphs, as well as to networks with different types of nodes, variously called multipartite, heterogeneous, or multimodal networks.

When permuting bipartite networks, our method ensures that each nodes class membership and with-class degree is preserved. Similarly, heterogeneous networks should be permuted by considering each edge type as a separate network. This way, each node retains its within-edge-type degree for all edge types. We provide documentation for parameter choices depending on the type of network being permuted in the GitHub repository (https://github.com/hetio/xswap). The original algorithm and our proposed modification are given in Figure 2.

A

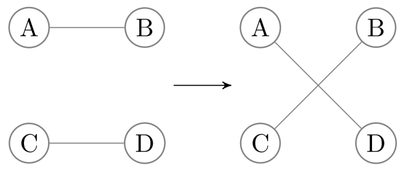
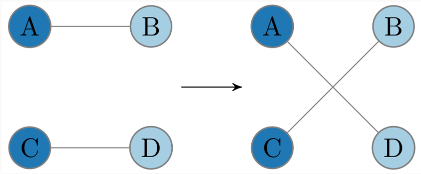
```
Input: Undirected graph G, distribution $\rho$, and number of steps $T$
Output: Edge-swapped graph $G_s$
  for i = 1,...,$T$ do
      Select two edges $(i, j)$, $(k, l) \in E(G_s)$
      if $(i, l) \notin E(G_s)$ and $(k, j) \notin E(G_s)$ then
          $E(\widehat{G_s}) \leftarrow \left(E(G_s) \setminus \{(i, j), (k, l)\}\right) \cup \{(i, l), (k, j)\}$
          $G_s \leftarrow \widehat{G_s}$ with probability $\min(\rho(\widehat{G_s})/\rho(G_s), 1)$
      end if
  end for
```

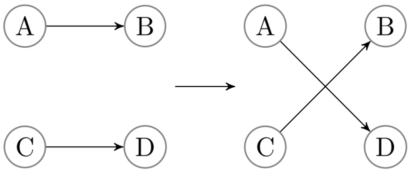
B

```
Input: Directed, undirected, or bipartite graph G, number of steps $T$, and
  booleans allow_antiparallel and allow_loops
Output: Edge-swapped graph $G_T$
  Initialize: $G_0 \leftarrow G$
  for i = 1,...,$T$ do
      Select two edges $(i, j)$, $(k, l) \in E(G_{i-1})$
      condition_1 $\leftarrow (i, l) \in E(G_{i-1})$ or $(k, j) \in E(G_{i-1})$
      condition_2 $\leftarrow$!allow_antiparallel and $((l, i) \in E(G_{i-1})$ or $(j, k) \in E(G_{i-1}))$
      condition_3 $\leftarrow$ !allow_loops and $(i \neq l$ or $k \neq j)$
      if condition_1 or condition_2 or condition_3 then
          continue $G_i \leftarrow G_{i-1}$
      else$E(G_i) \leftarrow (E(G_{i-1}) \setminus \{(i, j), (k, l)\}) \cup \{(i, l), (k, j)\}$
      end if
  end for
```

Figure 2: A. XSwap algorithm due to Hanhijärvi, et al. [14]. **B.** Proposed modification to XSwap algorithm

Table 1: Applications of the modified XSwap algorithm to various network types with appropriate parameter choices. For simple networks, each node's degree is preserved. For bipartite networks, each node's number of connections to the other part is preserved, and overall node class memberships are preserved. For directed networks, each nodes' in- and out-degrees are preserved, though parameter choices depend on the network being permuted. Some directed networks can include antiparallel edges or loops while others do not.

Network type	Degree preserved	Figure	`allow_antiparallel`	`allow_loops`
simple	all		False	False
bipartite	in/out		True	True

Network type	Degree preserved	Figure	`allow_antiparallel`	`allow_loops`
directed	in/out		?	?

Edge prior

We introduce the "edge prior" to quantify the probability that two nodes are connected based only on their degree. The edge prior can be estimated using the fraction of permuted networks in which a given edge exists—the maximum likelihood estimate for the binomial distribution success probability. Based only on permuted networks, the edge prior does not contain any information about the true edges in the (unpermuted) network. The edge prior is a numerical feature that can be computed for every pair of nodes that could potentially share an edge, and we compared its ability to predict edges in three tasks, discussed below.

Edge prior analytical approximation

We also considered the possibility that the probability of an edge existing across permuted networks could be written as a closed form equation involving the node pair's degree. A major simplification is the assumption that the probability of an edge existing is independent of all other potential edges. We were unable to find a closed-form solution giving the edge prior without assuming independence in this way, which we believe is incorrect for XSwap. Nonetheless, we discovered a good analytical approximation to the edge prior for networks with many nodes and relatively low edge density.

Let m be the total number of edges in the network, and $d(u_i)$, $d(v_j)$ be the source and target degrees of a node pair, respectively. A good approximation of the edge prior is given by the following:

$$P_{i,j} = \frac{d(u_i)d(v_j)}{\sqrt{(d(u_i)d(v_j))^2 + (m - d(u_i) - d(v_j) + 1)^2}}$$

Further discussion of this approximate edge prior and an derivation are available in the supplement.

Prediction tasks

Degree-grouping

Our method for degree-preserving permutation produces randomized networks that share few of their edges with the original network. The feature values for two node pairs with the same source and target degree are drawn from the same distribution in permuted networks, so nodes with equal degree can be grouped when summarizing features. We used this to augment each node pair's feature values in permuted networks, which allowed these pairs to have more permuted feature values than permuted networks. Degree grouping greatly increased the effective number of permutations for nodes with frequently observed degrees [15]. We used degree grouping throughout our analyses.

Implementation and source code

We implemented the modified XSwap algorithm as a Python package, with the actual edge swap mechanism implemented in C++ for greater speed. In addition to functions that permute networks (represented as edge lists), the package contains utilities for computing the edge prior, converting a network between adjacency matrix and edge list formats, and for assigning unique identifiers to nodes. The Python package is available on the Python Packaging Index under the name “xswap”. The full source code for our method of degree-preserving network permutation has also been made freely available (https://github.com/hetio/xswap), as has the code for the analysis, figure generation (https://github.com/greenelab/xswap-analysis), and manuscript (https://github.com/greenelab/xswap-manuscript).

Results

Figure 3: Discrimination figure

Figure 3: Discrimination figure

Figure 4: Calibration figure

Figure 4: Calibration figure

The second prediction task mirrored the first task, but it involved reconstructing networks based on subsampled networks with only 70% of the original edges. Because edges were sampled uniformly without replacement, the subsampled networks share similar degree distributions to the original networks (see Figure {[**???**]}B). Unlike in the first task, edges that were present in the sampled network were not tested and therefore are not included in the performance metrics. The results of the second prediction task further demonstrate a high level of performance for degree-sequence-based node pair features (Figure 3). The edge prior was able to reconstruct the unsampled network with an AUROC of greater than 0.9 in 14 of 20 networks. As was observed in the first task, node pair features computed in second prediction task were highly rank-correlated, meaning the AUROC values for different features were similar. While performance was slightly lower in the second task than the first, we found that many networks were still well reconstructed. The edge prior was the best calibrated feature for both tasks.

Discussion

Conclusion

References

1. **Systematic integration of biomedical knowledge prioritizes drugs for repurposing**
Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, Sergio E Baranzini
eLife (2017-09-22) https://doi.org/cdfk
DOI: 10.7554/elife.26726 · PMID: 28936969 · PMCID: PMC5640425

2. **Link Prediction Methods and Their Accuracy for Different Social Networks and Network Metrics**
Fei Gao, Katarzyna Musial, Colin Cooper, Sophia Tsoka
Scientific Programming (2015) https://doi.org/f7hvd9
DOI: 10.1155/2015/172879

3. **Bias tradeoffs in the creation and analysis of protein–protein interaction networks**
Jesse Gillis, Sara Ballouz, Paul Pavlidis
Journal of Proteomics (2014-04) https://doi.org/f3mn5f
DOI: 10.1016/j.jprot.2014.01.020 · PMID: 24480284 · PMCID: PMC3972268

4. **Correcting for the study bias associated with protein–protein interaction measurements reveals differences between protein degree distributions from different cancer types**
Martin H. Schaefer, Luis Serrano, Miguel A. Andrade-Navarro
Frontiers in Genetics (2015-08-04) https://doi.org/gf5t46
DOI: 10.3389/fgene.2015.00260 · PMID: 26300911 · PMCID: PMC4523822

5. **Effect of sampling on topology predictions of protein-protein interaction networks**
Jing-Dong J Han, Denis Dupuy, Nicolas Bertin, Michael E Cusick, Marc Vidal
Nature Biotechnology (2005-07) https://doi.org/dj5cm8
DOI: 10.1038/nbt1116 · PMID: 16003372

6. **Towards a proteome-scale map of the human protein–protein interaction network**
Jean-François Rual, Kavitha Venkatesan, Tong Hao, Tomoko Hirozane-Kishikawa, Amélie Dricot, Ning Li, Gabriel F. Berriz, Francis D. Gibbons, Matija Dreze, Nono Ayivi-Guedehoussou, … Marc Vidal
Nature (2005-09-28) https://doi.org/dw6q23
DOI: 10.1038/nature04209 · PMID: 16189514

7. **A Proteome-Scale Map of the Human Interactome Network**
Thomas Rolland, Murat Taşan, Benoit Charloteaux, Samuel J. Pevzner, Quan Zhong, Nidhi Sahni, Song Yi, Irma Lemmens, Celia Fontanillo, Roberto Mosca, … Marc Vidal
Cell (2014-11) https://doi.org/f3mn6x
DOI: 10.1016/j.cell.2014.10.050 · PMID: 25416956 · PMCID: PMC4266588

8. **The Impact of Multifunctional Genes on “Guilt by Association” Analysis**
Jesse Gillis, Paul Pavlidis
PLoS ONE (2011-02-18) https://doi.org/bs9
DOI: 10.1371/journal.pone.0017258 · PMID: 21364756 · PMCID: PMC3041792

9. **Biology, Methodology or Chance? The Degree Distributions of Bipartite Ecological Networks**
Richard J. Williams
PLoS ONE (2011-03-03) https://doi.org/fmtk6x
DOI: 10.1371/journal.pone.0017645 · PMID: 21390231 · PMCID: PMC3048397

10. **The Degree Distribution of Networks: Statistical Model Selection**
William P. Kelly, Piers J. Ingram, Michael P. H. Stumpf
Bacterial Molecular Networks (2011-10-28) https://doi.org/ddx5rx
DOI: 10.1007/978-1-61779-361-5_13 · PMID: 22144157

11. **Scale-free networks are rare**
Anna D. Broido, Aaron Clauset
Nature Communications (2019-03-04) https://doi.org/gfztz9
DOI: 10.1038/s41467-019-08746-5 · PMID: 30833554 · PMCID: PMC6399239

12. **Addressing false discoveries in network inference**
Tobias Petri, Stefan Altmann, Ludwig Geistlinger, Ralf Zimmer, Robert Küffner
Bioinformatics (2015-04-24) https://doi.org/f7rwgt
DOI: 10.1093/bioinformatics/btv215 · PMID: 25910697

13. **Evidence of probabilistic behaviour in protein interaction networks**
Joseph Ivanic, Anders Wallqvist, Jaques Reifman
BMC Systems Biology (2008) https://doi.org/dsz4kn
DOI: 10.1186/1752-0509-2-11 · PMID: 18237403 · PMCID: PMC2267158

14. **Randomization Techniques for Graphs**
Sami Hanhijärvi, Gemma C. Garriga, Kai Puolamäki
Proceedings of the 2009 SIAM International Conference on Data Mining (2009-04-30)
https://doi.org/f3mn58
DOI: 10.1137/1.9781611972795.67

15. **Degree-grouped permtuations by zietzm · Pull Request #96 · greenelab/hetmech**
GitHub
https://github.com/greenelab/hetmech/pull/96

Supplemental information

Approximate edge prior

To approximate the edge prior, we began by making two simplifications. First, we assumed independence between node pairs. This assumption does not actually hold for the XSwap algorithm, though it is a reasonable simplification for large, sparse networks. Second, we assumed that the XSwap process is stationary. This assumption also does not actually hold, but it was made because it significantly simplifies the problem. A single node pair has two possible states, “edge” and “no edge”. These states are not transient, and they are not periodic so long as more than one possible swap exists in the network. In almost all cases, then, our simplified model of the algorithm gives the state of a node pair as an ergodic process, independent of other node pairs.

Let $A_{i,j}$ represent the existence of edge (i, j) For a given node pair, (i, j), then, let $q_{i,j}$ represent the transition probability from the “no edge” state to the “edge” state in one successful iteration of the XSwap algorithm. Let $r_{i,j}$ represent the probability of the opposite transition (“edge” to “no edge”) in one successful iteration. With “no edge” represented as $[1, 0]^T$ and “edge” represented as $[0, 1]^T$, the transition matrix, P, is given by the following:

$$P^T = \begin{bmatrix} 1-q & r \\ q & 1-r \end{bmatrix}$$

The stationary distribution of this system should correspond to the distribution when the number of swaps goes to infinity. It can be found by computing the eigenvectors of the system, as we know that the stationary distribution vector, $\mathbf{v}$ satisfies $P^T\mathbf{v} = \mathbf{v}$. The normalized eigenvector $\mathbf{v}$ is given by

$$\mathbf{v} = \frac{1}{r/q+1}\begin{bmatrix} r/q \\ 1 \end{bmatrix}$$

The asymptotic edge probability is therefore

$$\frac{1}{r/q+1}.$$

Since node pairs are being treated as independent, the probability of an edge being created in one successful iteration, given that the edge does not currently exist, is the ratio of the number of edge choices involving nodes i and j to the total number of possible swaps, S. Let $d(u_i)$ represent the degree of source node i and $d(v_j)$ represent the degree of target node j.

$$q_{i,j} = \frac{d(u_i)d(v_j)}{S}$$

Similarly, the probability of an edge being eliminated in one iteration is the ratio of the number of edge choices involving (i, j) and any other valid edge to the total number of possible swaps. Let m be the total number of edges in the network.

$$r_{i,j} = \frac{m - d(u_i) - d(v_j) + 1}{S}$$

The approximate edge prior is, therefore,

$$\frac{d(u_i)d(v_j)}{m - d(u_i) - d(v_j) + 1 + d(u_i)d(v_j)}.$$

Unfortunately, we found that the above edge prior approximation is a poor approximation in many cases. We found that the following modified form (introduced in Methods) affords a superior approximation:

$$P_{i,j} = \frac{d(u_i)d(v_j)}{\sqrt{(d(u_i)d(v_j))^2 + (m - d(u_i) - d(v_j) + 1)^2}}$$

Because the modified form of the approximation offers a much superior fit to the data, we chose to include only the modified version in the Python package released, and we used only the modified form throughout our analysis.