SCIENTIFIC REPORTS
nature research

OPEN

A Bootstrap Method for Goodness of Fit and Model Selection with a Single Observed Network

Sixing Chen & Jukka-Pekka Onnela*

Network models are applied in numerous domains where data arise from systems of interactions among pairs of actors. Both statistical and mechanistic network models are increasingly capable of capturing various dependencies among these actors. Yet, these dependencies pose statistical challenges for analyzing such data, especially when the data set comprises only a single observation of one network, often leading to intractable likelihoods regardless of the modeling paradigm and limiting the application of existing statistical methods for networks. We explore a subsampling bootstrap procedure to serve as the basis for goodness of fit and model selection with a single observed network that circumvents the intractability of such likelihoods. Our approach is based on flexible resampling distributions formed from the single observed network, allowing for more nuanced and higher dimensional comparisons than point estimates of quantities of interest. We include worked examples for model selection, with simulation, and assessment of goodness of fit, with duplication-divergence model fits for yeast (S.cerevisiae) protein-protein interaction data from the literature. The proposed approach produces a flexible resampling distribution that can be based on any network statistics of one's choosing and can be employed for both statistical and mechanistic network models.

Networks are used to represent data from systems composed of interactions among pairs of actors (represented by nodes)[1–5]. Often in such systems, these interactions (represented by edges) can depend on the state of the rest of the system, such as other edges or attributes of nodes. One prominent example of this is triadic closure in social networks, where two people are more likely to be friends should they share a mutual friend[6]. While innovations in network models are increasing the capability to account for various dependencies in the data, this rich level of interconnectedness poses a problem for statistical methods for networks.

In typical statistical settings, the premise is that the data is composed of a collection of independent observations. Typical methods derive efficiency gains and consistency from a large number of samples due to this independence. However, in the network context where the structure of the network is of primary interest, the edges and their placement are the outcome of interest, but there are often multiple layers of dependence. Thus, the premise of independent observations is not met and most statistical methods are not applicable.

To better understand the limitations, we inspect two prominent paradigms of network models. First, statistical models are probabilistic models that specify the likelihood of observing any given network[7–9]. One example of these models is the family of exponential random graph models (ERGMs)[4], which uses observable network configurations (such as triangles and k-stars) as the natural sufficient statistics. Although popular in practice, ERGMs can be difficult to fit and to sample from, and they may not scale well to large networks[10]. Estimation of ERGMs is done using maximum pseudolikelihood estimation (MPLE)[11] or Markov chain Monte Carlo maximum likelihood estimation (MCMC-MLE)[12,13]. Pseudolikelihood methods for inference with ERGMs can lead to biased results due to the ignored dependence[14], while inference for MCMC-MLE proceeds via simulation from estimated model[13] and is thus entirely model based. Second, mechanistic models are composed of generative mechanisms that prescribe the growth and change of a network[15–20]. While they are easy to sample from, a mechanistic model allows for numerous paths that can be taken in the state space to produce any one observed network, making the likelihood (of all but the most trivial models) intractable. As a result, performing statistical procedures is difficult for such models and there is little existing work in the literature for doing so.

In situations where likelihood based methods are not available, one often resorts to resampling methods, such as bootstrap, jackknife, and permutation tests[21–23]. Although the different resampling methods operate differently,

Department of Biostatistics, Harvard T.H. Chan School of Public Health, 655 Huntington Ave, SPH2, 4th Floor, Boston, MA, 02115, USA. *email: onnela@hsph.harvard.edu

they all serve to create new data sets from a single observed data set that mimic the behavior of the original one to serve as a basis for statistical procedures. This is an attractive option for networks, especially if there is only a single observed network, such as the Internet or the World Wide Web. Having multiple resampled networks that resemble, in some ways, the original observed network allows one to bypass the problem of unwieldy or intractable likelihoods. Even in the best case of the likelihood having a simple functional form, the normalizing constants of ERGMs are generally unobtainable even for a network of modest size, since they require summing over a computationally infeasible number of possible network realizations. For example, in a network of m edges, one may need to consider all the possible $m!$ orders of adding the edges since the mechanisms of growth may depend on the existing state of the network.

In this paper, we explore using a resampling procedure as a basis for statistical procedures for a single observed network. While there is some existing research on resampling methods in network settings, our approach is distinct in many ways. First, there are methods for assessing the goodness of fit for a fitted model[24,25]. These methods generally work by drawing network realizations from the fitted model, and then assessing fit by comparing the value of a set of network statistics for the observed network to the distribution of these statistics in the generated draws. This resampling scheme is akin to that of the parametric bootstrap. Note that this can be done for the point estimate of individual statistics or those of multiple statistics simultaneously, e.g., functionals of the degree distribution. However, the resamples in these methods are only representative of the *fitted model* and not necessarily of the *observed network*, and comparisons are made based only on point estimates. Second, there are methods for a setting where there are multiple independent networks observed for maximum pseudolikelihood estimation (MPLE)[26]. This is similar to the typical statistical setting with multiple independent observations and is not applicable to the setting with a single observed network. Third, there are resampling methods based on subgraphs of subsamples of nodes in the observed network[27–31]. Ohara *et al.*[27], Bhattacharyya *et al.*[28], Thompson *et al.*[30], and Gel *et al.*[31] are aimed at estimation and uncertainty quantification of network centrality, distribution of subgraphs, and functionals of the degree distribution, while Ali *et al.*[29] is a subgraph-based method for network comparisons.

Our procedure makes use of the bootstrap subsampling scheme from Bhattacharyya *et al.*[28]. Importantly, our method addresses goodness of fit and model selection rather than estimation, and is based on the entire resampling distribution (rather than point estimates) of any set of statistics obtained from the sampled subgraphs. The flexible choice of network statistics allows an investigator to focus the criterion for model fit based on scientific interest. The full resampling distribution also contains more information than aggregated subgraph counts and point estimates for comparison with candidate models. The procedure also allows for natural uncertainty quantification regardless of the algorithm used for selecting the model, is agnostic to the modeling paradigm (statistical or mechanistic), and can accommodate any model one can sample from. The scaling of the procedure depends on the statistics chosen and the number of subsamples taken, where the latter scales linearly.

The rest of the paper is organized as the following. In Materials and Methods, we explain the proposed bootstrap subsampling procedure, highlight important considerations for some of the steps, and elaborate on potential use cases. In Results, some of the proposed use cases (model selection and goodness of fit) are demonstrated with simulated and empirical data. Lastly, we conclude with Discussion.

Materials and Methods

Subsampling scheme and resampling distributions. Each subsample of the bootstrap subsampling scheme of Bhattacharyya *et al.*[28] consists of a uniform node-wise subsample of all the nodes in the observed network G_o (with node set V_o and edge set E_o) and their induced subgraph, i.e., the nodes in the subsample and all edges between these nodes. For each subsample, one may compute any set of statistics to form a resampling distribution of these statistics. Although the subsamples will not have the same properties as the full network or a network of the same size as the subsample drawn from the true data generating mechanism[32], they will still retain features of the true data generating mechanism since the subsampling does not change any between-edge or between-node dependence that influenced the formation of the network, despite adding a degree of "missingness" by removing elements correlated with those in the subsample. In comparison, should one generate draws from a particular fitted model to form a resampling distribution, the between-edge and between-node dependencies will be those specified by the fitted model; in this case, the generated networks will only be representative of the true data generating mechanism if the fitted model is the true model, which is a strong assumption and usually not verifiable in practice.

Because each subsample only consists of a subset of V_o and E_o, each subsample will be missing elements that are correlated with those that are included in the subsample. As a result, this must be taken into account when any comparisons are made with a null/candidate model M_c. One may be tempted to compare subsamples of G_o with draws from M_c of the same size as the subsamples. This should however be avoided since there is a degree of "missingness" in the subsamples of G_o that are not present in such draws from M_c. Even if M_c was the true model, this disparity could make the two behave differently. Instead, one should generate draws from M_c of the same size as G_o and then apply the same subsampling scheme to these draws. This ensures that both the subsamples of G_o and those of M_c have the same amount of "missingness" and are comparable. Should M_c be representative of the true data generating mechanism, then the behavior of the two subsamples their corresponding resampling distributions of computed statistics should be similar. The representativeness of the subsamples from G_o, as well as their comparability with the subsamples from M_c, form the basis of our procedure. Even though we only consider uniform subsampling, the subsampling method is flexible and can be chosen to be representative of sampling in practice or for statistical and computational ease. The proposed bootstrap subsampling procedure is summarized in Fig. 1.

In contrast to existing methods that also use draws from the fitted model to assess goodness of fit, this approach can lead to a richer comparison. For existing methods, after choosing the statistics for assessing goodness of fit, the given statistics are computed for G_o and for a large number of draws from M_c. The point estimate of these statistics for G_o are then placed within the distribution of said statistics of the draws from M_c. Goodness

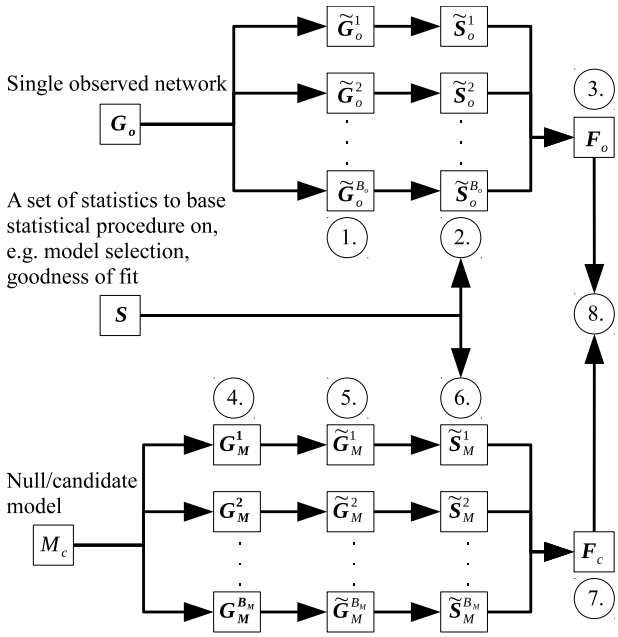

Figure 1. Schematic of the steps of the proposed bootstrap subsampling procedure for a single observed network G_o: 1. Obtain subgraphs induced by B_o subsamples of the nodes of $\mathbf{G}_o$. 2. Compute the chosen network statistics $\mathbf{S}$ for each induced subgraph. 3. Form resampling distribution F_o of $\mathbf{S}$ from $\widetilde{\mathbf{S}}_o^1 \ldots \widetilde{\mathbf{S}}_o^{B_o}$. 4. Draw networks of same size as $\mathbf{G}_o$ from network model M_c. 5. For each generated network $\mathbf{G}_M{}^1 \ldots \mathbf{G}_M^B M$, obtain one subgraph induced by one subsample of the nodes of each network. 6. Compute $\mathbf{S}$ for each induced subgraph ($\widetilde{\mathbf{S}}_M^1 \ldots \widetilde{\mathbf{S}}_M^{B_M}$). 7. Form resampling distribution F_c of $\mathbf{S}$ from $\widetilde{\mathbf{S}}_M^1 \ldots \widetilde{\mathbf{S}}_M^{B_M}$. 8. Perform statistical procedure by comparing F_o and F_c.

of fit is assessed by the location of the point estimate from G_o within the draws from M_c. This can be done visually or by quantifying the proportion of the draws with values of the statistics deemed more extreme. With our approach, the two resampling distributions can be compared in many ways, such as their location, spread, and shape. In addition, one can quantify the distance between the two distributions using, for example, the Kolmogorov-Smirnov (KS) statistic (defined for discrete distributions also[33]) or the Kullback-Leibler divergence to order the fit of different candidate models.

One point of interest and emphasis is that the subsamples from G_o are all from a single network, while the subsamples from M_c are subsamples of independent network realizations drawn from M_c instead of subsamples from a single network drawn from M_c. This scheme is proposed due to potential instability of single generated networks and the corresponding subsamples, since there can be a great deal of instability in the generated networks depending on the model, including the seed network used to grow networks specified by mechanistic models. In addition, the disparity between the two types of subsamples may depend on the proportion of the nodes in each subsample. Both of these points are important to the performance of the procedure and are further examined in the next two sections.

Stability under sampling. When sampling from the candidate model, one needs to take care to ensure that the draws from M_c behave like the observed network even if the candidate model is the true model or an accurate model, and in turn, the subsamples of these draws behave like the subsamples of the observed network. In the worst case, such draws can look nothing like the observed network despite using a good candidate model, e.g., the draws could have highly varying degree distributions that look nothing like that of the observed network. This issue can be more prominently demonstrated in the context of some mechanistic network models.

Networks generated from mechanistic models are often grown from a small (relative to the final size of the network) seed network according to the model's generative mechanism until some stopping condition is reached, e.g., attaining a requisite number of nodes. There is research showing that the original seed network has no influence on the degree distribution in the limit, i.e., for a large number of nodes, for certain types of mechanistic network models[34,35]. While some data sets, such as online social networks, may be sufficiently large to reach this asymptotic regime, others, such as protein-protein interaction networks, may not be. Thus, when generating draws from candidate models for analysis of smaller networks, the original seed network can potentially have a great deal of influence. The seed network maybe as simple as a single node, or a complete graph of only three nodes, up to bigger complete graphs, or something more elaborate with more than one component. We briefly examine the effect of the seed network on the stability of the degree distribution of networks generated from the Erdös-Rènyi and duplication-divergence models, which are frequently used to model protein-protein interaction networks.

Erdös-rènyi model. The Erdös-Rènyi (ER) model[36] is a simple but rather unique model in that it can be framed as both a mechanistic and a statistical model. In the ER model, the number of nodes n is fixed, and there are two

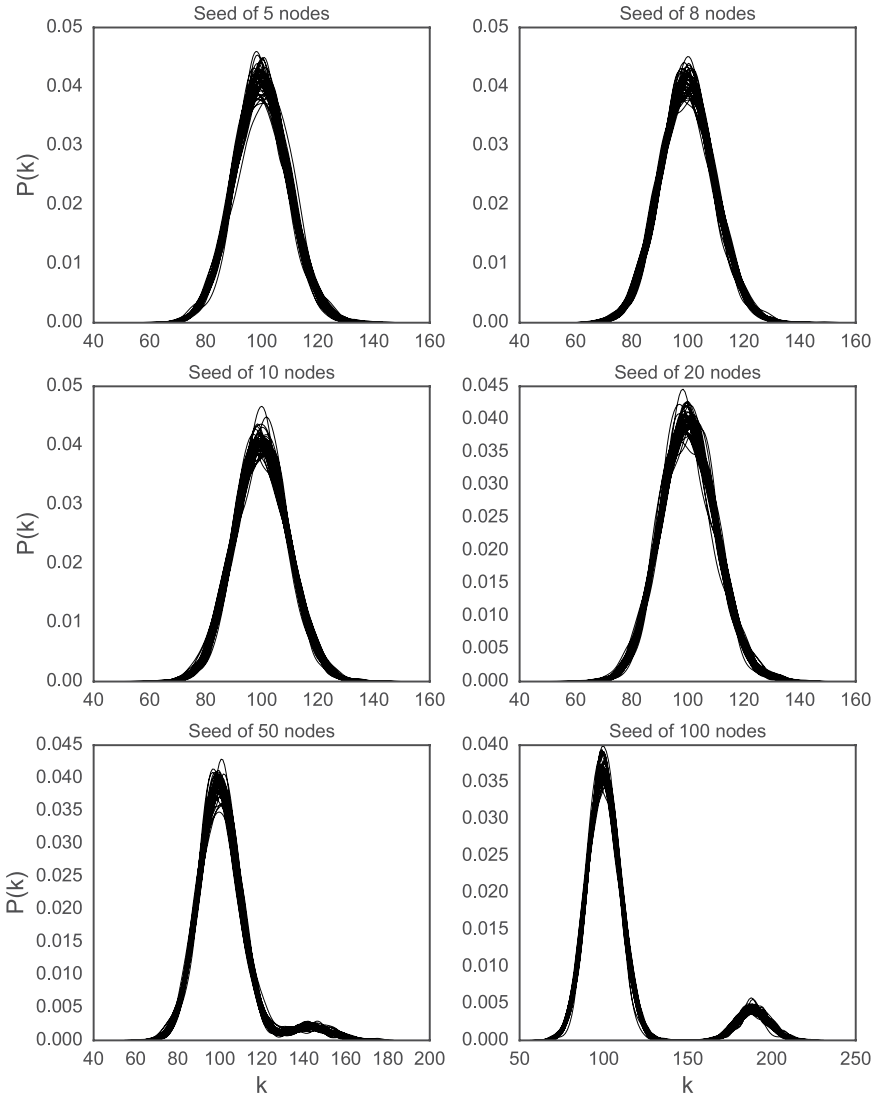

Figure 2. The degree distribution of 50 generated graphs from the $G(n = 1000, p = 0.1)$ model with seeds of 5, 8, 10, 20, 50, 100 nodes, from left to right, then top to bottom, as described in text. This and the next figure show the differing influence the seed can have on variability.

variants of the model that determine how the edges are placed. In the first variant, the $G(n,p)$ model, each of the $C(n, 2)$, n choose 2, possible edges are independent and are included in the graph with probability p, so the number of edges in the graph is binomial. In the other variant, the $G(n,m)$ model, the number of edges in the graph m is also fixed. In this case, the random graph has a uniform distribution over all $C(C(n, 2),m)$ possible graphs with n nodes and m edges.

The first variant can be easily framed as a mechanistic model. The network generation starts with a seed network of a single node. Then at each stage, a new node is added, and an edge between the new node and each existing node is added with probability p. This is done until there are n nodes in the network. Rather than starting with a seed network of a single node, networks can be generated according to the generative mechanism of the $G(n,p)$ model initialized with a different seed network. Here, we generated $G(n = 1000, p = 0.1)$ networks according to these rules, with complete graphs of 5, 8, 10, 20, 50, 100 nodes as the seed networks. We generated 50 networks of each size of the seed to evaluate the influence of the seed network on the stability of the degree distribution of the fully grown network.

The degree distribution of the 50 generated graphs at each size of the seed network are plotted in Fig. 2. While the shape of the degree distribution understandably changes as the complete graph used as the seed network gets bigger, the size of the seed network seems to have little influence on the stability of the degree distribution. All 50 networks, for each size of the seed network, have very similar degree distributions. The width of the "band" of the 50 distributions stacked on top of one another also looks to be mostly unchanging. This seems to indicate that the variability in the degree distribution is largely unaffected by the size of the seed network.

Duplication-divergence models. Duplication-divergence models are a popular class of models used for protein-protein interaction networks. Some examples include the duplication-mutation-complementation

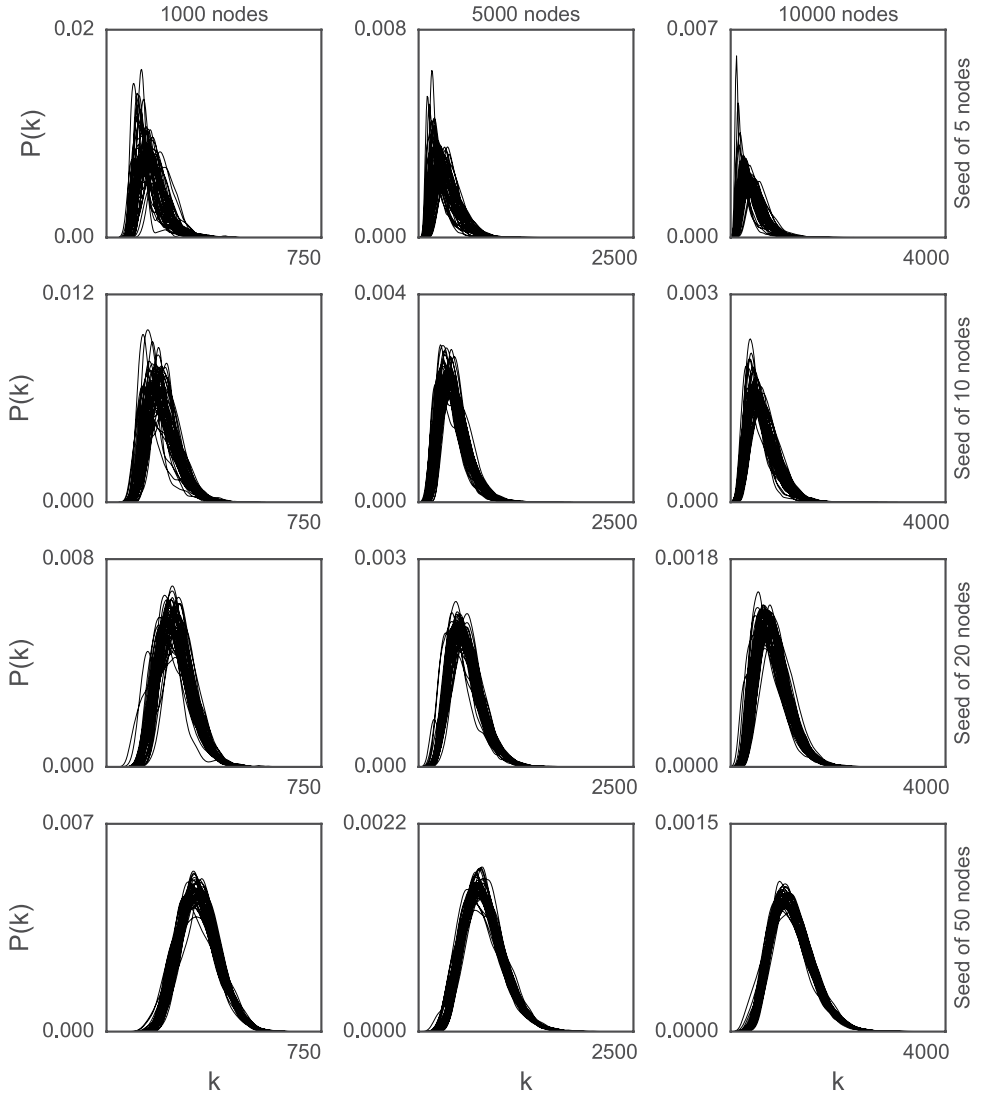

Figure 3. Degree distribution of 50 generated graphs of 1000, 5000, 10000 nodes from the DMC model, left to right, with seeds of 5, 10, 20, 50 nodes, top to bottom. This and the previous figure show the differing influence the seed can have on variability.

(DMC)[18] and duplication-mutation-random mutation models (DMR)[17,37]. Given a seed network, both DMC and DMR models grow the network according to their respective generative mechanisms until the requisite number of nodes, n, is reached. In both the DMC and DMR models, a new node is first added at the beginning of each step in network generation. An existing node is chosen uniformly at random for duplication, and an edge is then added between the new node and each neighbor of the chosen node. After this, the two models diverge. For DMC, for each neighbor of the chosen node, one of the edge between the chosen node and the neighbor or the edge between the new node and the neighbor is randomly chosen and then removed with probability q_{mod}. The step is concluded by adding an edge between the chosen node and the new node with probability q_{con}. For DMR, each edge connected to the new node is removed independently with probability q_{del}. The step concludes by adding an edge between the new node and any existing node at the start of step t with probability $q_{new}/n(t)$, where $n(t)$ is the number of nodes in the network at the start of step t.

To assess the stability of the degree distribution, we generated 50 network realizations of 1000, 3000, 5000, 7000, 10000 nodes from both models with the seed network set as a complete graph with 5, 8, 10, 20, 50, 100 nodes. The parameters of the DMC model were set as $q_{mod} = 0.2$ and $q_{con} = 0.1$, while those of the DMR model were $q_{del} = 0.2$ and $q_{new} = 0.1$. The degree distribution for the 50 generated DMC networks for a subset of all combinations of the size of the seed network and the total number of nodes are plotted in Fig. 3; those for all combinations for both DMC and DMR models can be found in the Supplementary Information (Figs S1 and S2). A general trend in the plots is that the total number of nodes in the network has little to no influence on the stability of the degree distribution, while the size of the seed network has a great deal of influence, with stability increasing sharply with the size of the seed network, up to 50. For smaller seed networks (5 nodes), the shape and spread of the degree distributions vary wildly even for larger networks. For a modest increase in the size of the

seed network (10 nodes), the shape and the spread of the degree distributions become more similar. Finally, for larger seed networks (20 or 50), the shape and spread of the degree distributions are quite uniform, and the width of the "band" of the 50 degree distributions stacked on top of one another also decreases. Clearly, the variability of the degree distribution depends greatly on the size of the seed network.

One important difference between the ER and DMC/DMR models is the dependence on exisiting edges on the formation of new ones. The instability in the degree distribution of networks generated from DMC/DMR models with small seed networks can be attributed to this dependence. While these two examples show the influence the seed network can potentially have in generating networks of modest size with mechanistic models, it does beg the question of how one selects a meaningful seed that leads to stable sampling while mimicking the behavior of the observed network in a principled way. Hypothetically, if the observed network is indeed generated from an ER model and assuming the seed network and the parameter values are well chosen, then the generated networks should mostly appear similar to the observed network due to the low variability regardless of the size of the seed. On the other hand, should the observed network come from a DMC/DMR model and assuming well chosen parameter values, as well as an appropriate but small seed network, then the generated networks are unlikely to appear similar to the observed network due to the high variability with small seeds as demonstrated.

Portion of nodes to include in subsamples. The portion of nodes included in each subsample should not be so small such that no characteristics of the observed network or candidate models are retained, but also not so big such that the subsamples contain little variability. In one extreme, each subsample consists of just one node so that there is no structure within the induced subgraph, and in the other extreme, each subsample is simply the entire network. While the latter is of little concern when taking subsamples from independent draws from candidate models, it leaves no variability in the subsamples from a single observed network such that any resulting resampling distribution would simply be a point mass. We investigate what is an appropriate portion of nodes to include in each subsample through a detailed example with one particular model. The details can be found in the Supplementary Information.

In our example, we define the criterion for performance in terms of the expectation of the KS statistic (smaller values are better) between F_1, the resampling distribution from the subsamples of a single network drawn from candidate model M_c, and F_c, the resampling distribution from subsamples of several independent networks drawn from M_c, where each subsample comes from a different independent draw. This quantity is a measure of how closely F_o, the resampling distribution from the subsamples of the observed network, matches F_c when the observed network is truly generated by M_c. If the KS statistic is small, discrepency between F_o and F_c will be small if the model is correct. Additionally, this quantity being small implies that there is not much difference between using F_1 or F_c for comparison with F_o, thus we would be better off in electing for the stability of F_c. Note that the computation time required for F_c is greater than that for F_1. Although not completely generalizable, our example suggests to keep the portion of nodes in the subsample low ($<$30% in this example) as long as enough features of the models can be retained.

Proposed use cases. There are a variety of statistical procedures that can take advantage of this sampling scheme, with a few of them detailed below. Before proposing the general framework for a few typical statistical procedures via the bootstrap subsampling procedure, we define the following notation for the rest of the section. The observed network will be referred to as G_o with B_o subsamples and corresponding induced subgraphs $\widetilde{G}_o^{(1)} \dots \widetilde{G}_o^{(B_o)}$. The draws from candidate model M_c will be referred to as $G_M{}^1 \dots G_M^B M$ with corresponding subsample induced subgraphs $\widetilde{G}_M^{(1)} \dots \widetilde{G}_M^{(B_M)}$. Given a set of network statistics S chosen for model selection or assessing goodness of fit, the set computed from $\widetilde{G}_o^{(1)} \dots \widetilde{G}_o^{(B_o)}$ will be referred to as $\widetilde{S}_o^{(1)} \dots \widetilde{S}_o^{(B_o)}$, while those computed from $\widetilde{G}_M^{(1)} \dots \widetilde{G}_M^{(B_M)}$ will be referred to as $\widetilde{S}_M^{(1)} \dots \widetilde{S}_M^{(B_M)}$. Note that B_o and B_M need not be equal.

Model selection. Suppose the goal is to select between candidate models $M_1 \dots M_c$ for G_o. Given a set of statistics S to base the model selection on, one needs to compute $\widetilde{S}_{M_i}^{(1)} \dots \widetilde{S}_{M_i}^{(B_M)}$ from $\widetilde{G}_{M_i}^{(1)} \dots \widetilde{G}_{M_i}^{(B_M)}$ for $i = 1 \dots c$. These collections of statistics along with the model indices of each draw form the training data and are the basis for the model selection procedure. The selection of S is flexible and should be chosen to prioritize the aspects of the network where similarity to the observed network is most paramount. The training data can be used to train any learning algorithm for prediction of the model index. Examples include random forest, support vector machine, and ensemble learning algorithms like the Super Learner[38–40]. The trained algorithm is evaluated at each of $\tilde{S}_o^{(1)} \dots \tilde{S}_o^{(B_o)}$ to give selected model $\hat{M}_1 \dots \hat{M}_{B_o}$, with majority rule deciding the final selected model.

Algorithm 1. Steps for the model selection with the bootstrap subsampling procedure.

1. Draw subsamples $\tilde{G}_o^{(1)} \dots \tilde{G}_o^{(B_o)}$ from G_o
2. Draw subsamples $\tilde{G}_{M_i}^{(1)} \dots \tilde{G}_{M_i}^{(B_M)}$ from each candidate model $i = 1 \dots c$
3. Compute statistics for model selection for $\tilde{G}_o^{(1)} \dots \tilde{G}_o^{(B_o)}$ and $\tilde{G}_{M_i}^{(1)} \dots \tilde{G}_{M_i}^{(B_M)}$ for each $i = 1 \dots c$
4. Form training data based on each of $\tilde{S}_{M_i}^{(1)} \dots \tilde{S}_{M_i}^{(B_M)}$ along with model index i
5. Train a learning algorithm using training data: the predictors are the network statistics and the outcome is the model index i
6. Evaluate trained algorithm on $\tilde{S}_o^{(1)} \dots \tilde{S}_o^{(B_o)}$ and select the model based on plurality rule

One distinct advantage of the model selection through this bootstrap subsampling procedure is that it gives inherent evidence about uncertainty or confidence in the selected model as well as other candidate models. The proportion of $\widetilde{G}_o^{(1)} \ldots \widetilde{G}_o^{(B_o)}$ that are assigned to each model can be seen as evidence in favor of each candidate model, while the proportion of subsamples assigned the model that forms the majority can be seen as confidence in the selected model. With algorithms like random forest, where the decision is based on plurality rule, this aspect of our approach does not add anything new. But with others, such as support vector machine or the Super Learner that are not based on plurality rule, this approach offers a way to quantify uncertainty without the need to alter the learning algorithm itself.

Goodness of fit. To assess the goodness of fit for candidate models $M_1 \ldots M_c$, the procedure is similar to that of model selection. For a set of statistics S for assessing goodness of fit, one computes $\widetilde{S}_o^{(1)} \ldots \widetilde{S}_o^{(B_o)}$ from $\widetilde{G}_o^{(1)} \ldots \widetilde{G}_o^{(B_o)}$ and $\widetilde{S}_{M_i}^{(1)} \ldots \widetilde{S}_{M_i}^{(B_M)}$ from $\widetilde{G}_{M_i}^{(1)} \ldots \widetilde{G}_{M_i}^{(B_M)}$ for $i = 1 \ldots c$. Rather than training a learning algorithm based on $\widetilde{S}_{M_i}^{(1)} \ldots \widetilde{S}_{M_i}^{(B_M)}$ as in model selection, $\widetilde{S}_o^{(1)} \ldots \widetilde{S}_o^{(B_o)}$ are directly compared against $\widetilde{S}_{M_i}^{(1)} \ldots \widetilde{S}_{M_i}^{(B_M)}$ for each i to assess fit. As mentioned above, this comparison between the distribution of $\widetilde{S}_o^{(1)} \ldots \widetilde{S}_o^{(B_o)}$ and any set of $\widetilde{S}_{M_i}^{(1)} \ldots \widetilde{S}_{M_i}^{(B_M)}$ can be done in terms of location, spread, shape, or other aspects of the distribution.

Algorithm 2. Steps for assessing goodness of fit with the bootstrap subsampling procedure.

1. Draw subsamples $\tilde{G}_o^{(1)} \ldots \tilde{G}_o^{(B_o)}$ from G_o
2. Draw subsamples $\tilde{G}_{M_i}^{(1)} \ldots \tilde{G}_{M_i}^{(B_M)}$ from each candidate model $i = 1 \ldots c$
3. Compute $\tilde{S}_o^{(1)} \ldots \tilde{S}_o^{(B_o)}$ from $\tilde{G}_o^{(1)} \ldots \tilde{G}_o^{(B_o)}$, and $\tilde{S}_{M_i}^{(1)} \ldots \tilde{S}_{M_i}^{(B_M)}$ from $\tilde{G}_{M_i}^{(1)} \ldots \tilde{G}_{M_i}^{(B_M)}$
4. Assess fit by comparing $\tilde{S}_o^{(1)} \ldots \tilde{S}_o^{(B_o)}$ and $\tilde{S}_{M_i}^{(1)} \ldots \tilde{S}_{M_i}^{(B_M)}$ for each i

Assessment based on any one of these aspects may however lead to conflicting results, i.e., different models having the best fit depending on which aspect the comparison is based on, and it might be desirable to make comparisons through a more holistic measure. One solution to this is to compute a distance measure, such as the KS statistic or the Kullback-Leibler divergence, between $\widetilde{S}_o^{(1)} \ldots \widetilde{S}_o^{(B_o)}$ and $\widetilde{S}_{M_i}^{(1)} \ldots \widetilde{S}_{M_i}^{(B_M)}$ to quantify the fit of model i. This gives a single statistic that takes the entire distribution into account to quantify and to categorically order the fit of each candidate model. The KS test statistic and Kullback-Leibler divergence are typically computed in one dimension and can be used to compare the fit for each statistic individually as is. Instead, should one wish to make a comparison based on all statistics S at the same time, one can look to use generalizations of these statistics[41–43].

Comparison of multiple networks. If multiple networks are observed instead of a single network, and the goal is to assess how similar they are, then one can do so by building a resampling distribution from multiple networks. For the case of two observed networks with a set of statistics S for comparison and observed networks G_{o1} and G_{o2}, one can compute $\widetilde{S}_{o1}^{(1)} \ldots \widetilde{S}_{o1}^{(B_{o1})}$ and $\widetilde{S}_{o2}^{(1)} \ldots \widetilde{S}_{o2}^{(B_{o2})}$ from subsamples $\widetilde{G}_{o1}^{(1)} \ldots \widetilde{G}_{o1}^{(B_{o1})}$ and $\widetilde{G}_{o2}^{(1)} \ldots \widetilde{G}_{o2}^{(B_{o2})}$, respectively. The comparison of the two is based on $\widetilde{S}_{o1}^{(1)} \ldots \widetilde{S}_{o1}^{(B_{o1})}$ and $\widetilde{S}_{o2}^{(1)} \ldots \widetilde{S}_{o2}^{(B_{o2})}$, and one can proceed essentially the same way as with goodness of fit by comparing different aspects of the two distributions, but with $\widetilde{S}_{o1}^{(1)} \ldots \widetilde{S}_{o1}^{(B_{o1})}$ and $\widetilde{S}_{o2}^{(1)} \ldots \widetilde{S}_{o2}^{(B_{o2})}$ in place of $\widetilde{S}_o^{(1)} \ldots \widetilde{S}_o^{(B_o)}$ and $\widetilde{S}_{M_i}^{(1)} \ldots \widetilde{S}_{M_i}^{(B_M)}$. Should there be more than two observed networks for comparison, then the distance measure statistics can once again be used to quantify all pairwise relative similarities between the observed networks.

Results

Simulation and empirical data. We use simulation studies as well as data from an empirical network to illustrate the use of the bootstrap subsampling procedure in some of the scenarios described in the previous section. The simulated data and all code can be found under the Supplementary Information, while the protein-protein interaction data can be downloaded from the database of interacting proteins (DIP)[44] website directly.

Model selection. The simulation studies conducted for model selection consider instances of a variation on the aforementioned $G(n,m)$ model we introduced[40]. This variation generates random graphs with n nodes and m edges just as the $G(n,m)$ model with each edge being added one at a time. At each step in network generation, a pair of unconnected nodes are selected at random, and the probability for adding an edge between the two is determined based on the number of triangles it would close; the edge is then added with the given probability. This is repeated until there are m edges in the network. If the probability for adding an edge is fixed, then this is the $G(n,m)$ model. Instead, we start with a base probability p_0 to add the edge. Should the edge close at least one triangle, the probability increases by p_1. Should multiple triangles be closed by the edge, then the probability further increases by p_Δ for each additional triangle closed.

In the simulation, we select between two instances of this model, both having $p_0 = 0.3$ and $p_1 = 0.1$. The difference comes in p_Δ, with $p_\Delta = 0$ for model 1, while p_Δ varies over 0.05, 0.03, 0.01, 0.005 for model 2. For a given choice of n and m, as p_Δ decreases and gets closer to 0, the difference between the two models becomes more difficult to detect. The generated networks from both models consist of 100 nodes with edge count varying over

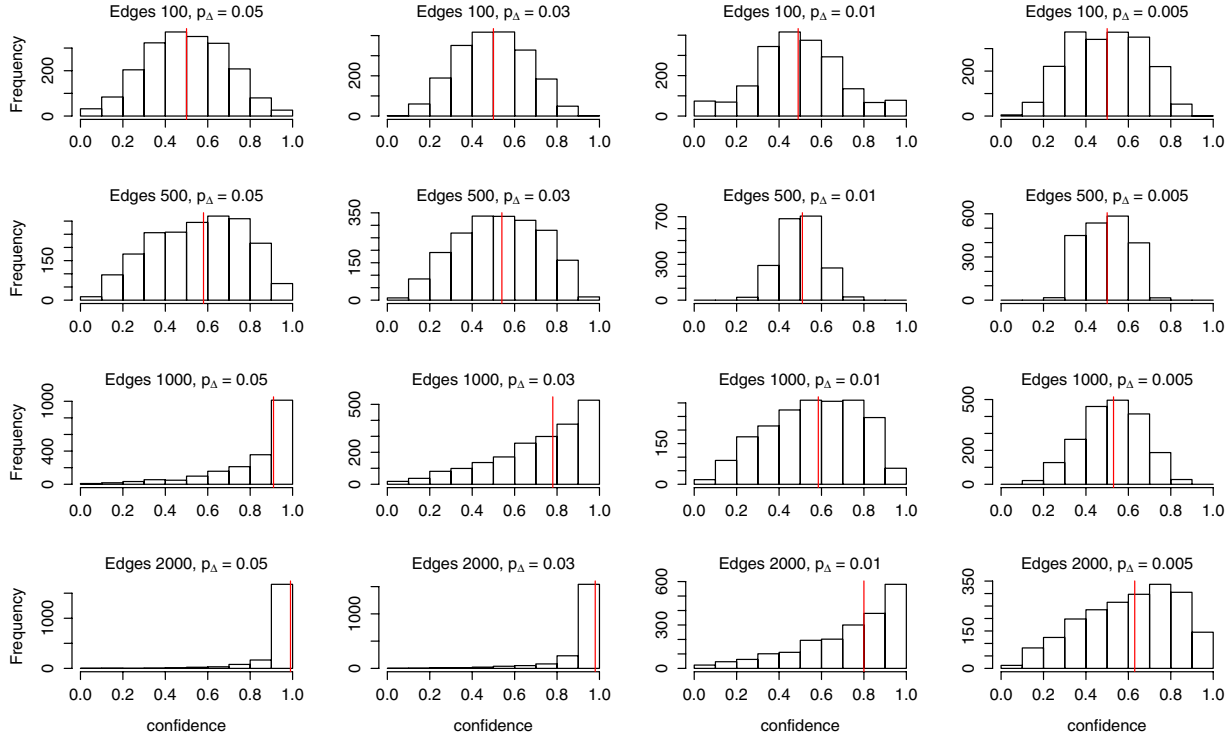

Figure 4. Histograms of the confidence score (proportion of subsamples assigned the correct model here rather than the majority) for p_Δ from 0.05, 0.03, 0.01, 0.005, from left to right, and edge count from 100, 500, 1000, 2000, from top to bottom, with the red vertical lines representing the median. This shows that our proposed approach for model selection behaves as one would intuitively expect, i.e., greater differences between the models are more frequently classified correctly than smaller differences.

	$p_\Delta = 0.05$	0.03	0.01	0.005
Edge count = 100	0.5015	0.5005	0.4834	0.5100
500	0.6092	0.5670	0.5178	0.5076
1000	0.9203	0.8202	0.6249	0.5786
2000	0.9890	0.9740	0.8343	0.6810

Table 1. Proportion of the test networks correctly classified at each combination of p_Δ and edge count.

100, 500, 1000, 2000. This gives a total of 20 comparisons between the models, one for each combination of values of p_Δ and m. For a given set of parameter values, the difference between the two models should be easier to detect as edge count increases, since the difference due to p_Δ has more opportunities to manifest itself. The training data consists of a single subsample of 80 nodes for each of 10000 draws from each model ($\widetilde{G}_{M_i}^{(1)} \ldots \widetilde{G}_{M_i}^{(10000)}$), where $i = 1$, 2. The test data consists of 1000 draws from each model (G_o), while the model selection is based on 100 subsamples of 80 nodes from each draw ($\widetilde{G}_o^{(1)} \ldots \widetilde{G}_o^{(100)}$). Although 100 nodes seems few, it is already large enough for a network to give rise to a very large resampling distribution. Additionally, despite the simplicity of the model we are using, 100 nodes is large enough for the likelihood function to be intractable.

The model selection is through the Super Learner[38–40], with support vector machine (ν-classification with $\nu = 0.5$, radial kernel), random forest ($N_{tree} = 1000$, min terminal node size = 1), and k-nearest neighbors ($k = 10$) as candidate algorithms, and average clustering coefficient, triangle count, and the three quartiles of the degree distribution as predictors. Note the parameters for the candidate algorithms are in parentheses. These statistics were chosen as predictors since the difference in p_Δ directly affects formation of triangles, while the other statistics are influenced strongly by triangles. For each of the 100 $\widetilde{G}_o^{(b_o)}$ for a particular testing network G_o, the Super Learner will give a score between 0 and 1 for predicting the model class of $\widetilde{G}_o^{(b_o)}$, with score < 0.5 assigned model 1 and score > 0.5 assigned model 2. The selected model is the model assigned to $\widetilde{G}_o^{(b_o)}$ s more frequently.

The results of the simulation are summarized in Fig. 4 and Table 1. Table 1 contains the proportion of test networks whose model was correctly classified by the Super Learner at each combination of p_Δ and edge count. Unsurprisingly, the proportion decreases as p_Δ decreases for a fixed edge count, and increases as edge count increases for a fixed p_Δ. Figure 4 shows the histogram of the confidence for the correct model. When model 1 is the true model of the test network, this is the proportion of the 100 subsamples that were assigned model 1, and

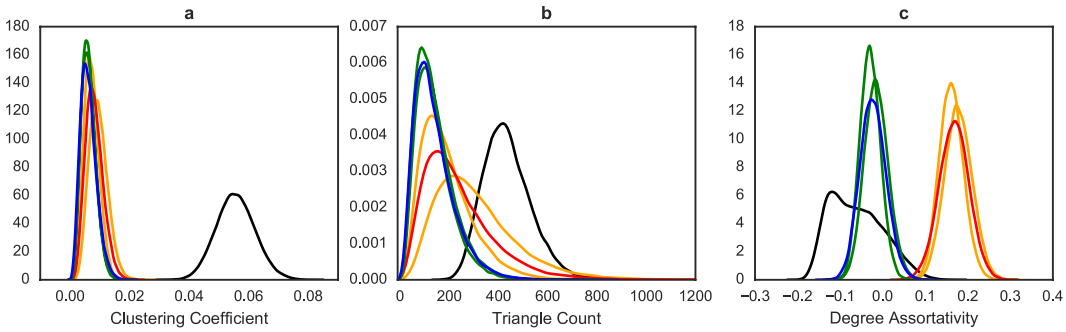

Figure 5. The *resampling* distribution of clustering coefficient (panel a), triangle count (panel b), and degree assortativity (panel c) from independent draws from the two model fits (blue for Hormozdiari *et al.*[45] and red is for Schweiger *et al.*[46]) as well as the PPI network (black). In addition, there are two resampling distributions from a single draw from each of the two model fits (green for Hormozdiari *et al.*[45] and orange is for Schweiger *et al.*[46]). This figure gives a visual representation of the additional information provided by the goodness of fit approach as well as difference from comparing point estimates with distribution of the statistics from full networks as seen in Fig. 6.

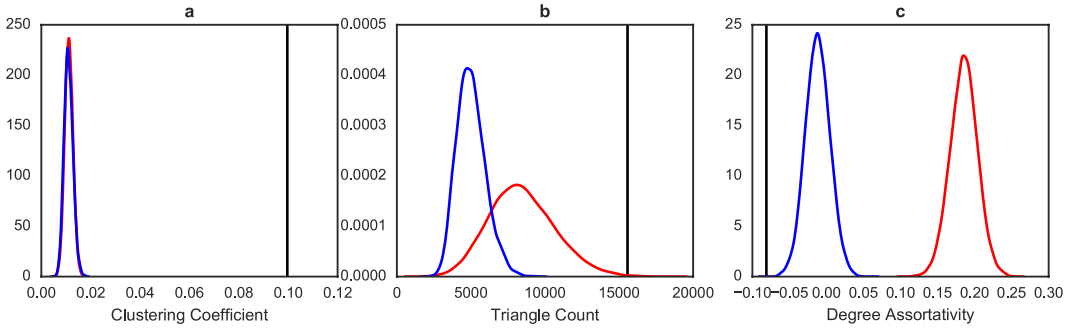

Figure 6. The distribution of clustering coefficient (panel a), triangle count (panel b), and degree assortativity (panel c) from independent *full network* draws from the two model fits (blue for Hormozdiari *et al.*[45] and red is for Schweiger *et al.*[46]). The corresponding point estimates from the *full* PPI network are the vertical black lines.

vice versa. When the proportion of correctly classified models is around 0.5, i.e., as good as a random guess, the confidence is symmetric and centered close to 0.5. When the proportion is higher than 0.5, the distribution of the confidence is shifted to the right, meaning that the two models are easier to tell apart. In addition, the more right skewed the histograms, the more confidence in the correct model. The red vertical line indicates the median, which also moves to the right as the proportion increases and as the confidence becomes more right skewed. This behavior indicates that the confidence for the selected model from the bootstrap subsampling procedure quantifies well the degree of uncertainty in the selected model. Random forest feature importance of all five predictors can be found in the Supplementary Information (Fig. S3) to see the shifting role of the predictors in the different scenarios.

Goodness of fit. To display our method for assessment of goodness of fit, we examine the yeast (*S.cerevisiae*) protein-protein interaction network data from DIP[44]. This data set has been much examined in the literature, including using network models. There are two particular publications[45,46] that fit different duplication divergence models to two different previous versions of the yeast data set, with differing seed networks. Here we apply our method to compare the fit of the two different models on the most recent version of the data.

Both papers use the same duplication divergence model[17,37], which we described as DMR earlier. However, the papers used different parameter values and different seed networks. The fit from Hormozdiari *et al.*[45] has parameter values $p = 0.365$ and $r = 0.12$, and the seed network contains 50 nodes. The seed network was constructed by highly connecting cliques, complete graphs where an edge exists between every pair of nodes, of 7 nodes and 10 nodes, then connecting additional nodes to the cliques. To highly connect the cliques, each possible edge between nodes in different cliques (70 such edges) was added with probability 0.67. Then, another 33 nodes were attached to randomly chosen nodes from the two cliques. At each step of the network generation, if a singleton (a node not connected to any other node) was generated, it was immediately removed in their model. Note that the details for obtaining the seed network from Hormozdiari *et al.*[45] were somewhat incomplete, so this is our interpretation of the description of their seed network.

On the other hand, the fit from Schweiger *et al.*[46] has parameter values $p = 0.3$ and $r = 1.05$. The authors use a smaller seed network of 40 nodes, generated with an inverse geometric model. To generate this seed network, a set of coordinates $\{x_1 \dots x_{40}\}$ in R^d is generated for each node. Then, each pair of nodes with distance $x_i - x_j$ greater than some threshold R is connected with an edge. Each dimension of the coordinates is independently generated from the standard normal distribution $N(0, 1)$. In their fit, the seed network uses $d = 2$ and $R = 1.5$. Unlike Hormozdiari *et al.*[45], Schweiger *et al.*[46] does not remove singletons as they are generated.

Both papers assessed the fit of their model by comparing certain aspects of the generated network to those of the yeast PPI network. In Hormozdiari *et al.*[45], model fit was assess via k-hop reachability, the number of distinct nodes reachable in $\leq k$ edges, the distribution of particular subraphs, such as triangles and stars, as well as some measures of centrality. Schweiger *et al.*[46] assess fit with the distribution of bicliques, i.e., subgraphs of two disjoint sets of nodes where every possible edge between the two sets exists. Here, we assess the fit of both models via our method with the average local clustering coefficient[16], triangle count, and the degree assortativity[47]. The local clustering coefficient of a particular node is a measure of to what extent its neighbors resemble a clique. Mathematically, this is computed as the number of edges between a node's neighbors divided by the maximum possible number of such edges. We use the average of the local clustering coefficient over all nodes in the network as a meassure of local clustering that is also attributable to the network as a whole. We also consider the number of triangle subgraphs that appear in the network. Unlike Hormozdiari *et al.*[45], which counts the total number of various subgraphs together, the count of triangles alone is a strictly global measure of clustering. Lastly, the degree assortativity of a network is a measure of how similar are the degrees of nodes connected by an edge. It is defined as the Pearson correlation of the degrees of nodes connected by an edge, so positively assorted networks have more edges between nodes of similar degrees, while negatively assorted networks have more edges between nodes of dissimilar degrees.

For the analysis, we consider the largest connected component (LCC) of the PPI network just as in Hormozdiari *et al.*[45]. The full network from the current version of the data contains 5176 nodes and 22977 edges, while the LCC contains 5106 (98.6%) nodes and 22935 (99.8%) edges. Networks drawn from each model contain the same number of nodes as the LCC, starting from their respective seed networks described above. Subsamples from the PPI network as well as networks drawn from each model contain 1550 nodes, roughly corresponding to 30%. This was the largest portion considered in our study of portion of nodes subsampled above.

The results of the data analysis are summarized in Fig. 5, where it is clear that the ordering of the fit of both models differs based on the network statistic of comparison. In accordance with earlier notation, for each statistic, we refer to the resampling distribution of the model of Hormozdiari *et al.*[45] as F_c^h and that of Schweiger *et al.*[46] as F_c^s, while that of the PPI network is referred to as F_o.

For clustering coefficient (Fig. 5a), both models fit equally poorly, as neither F_c^h nor F_c^s have any overlap with F_o. The KS statistic between F_o and each of F_c^h and F_c^s are both 1, indicating very poor fit. For triangle count (Fig. 5b), the model of Schweiger *et al.*[46] seems to fit better as F_c^s's spread has a much bigger overlap with F_o. The KS statistic between F_o and F_c^s (0.6778) is also much smaller than that between F_o and F_c^h (0.9018). Lastly, for degree assortativity (Fig. 5c), the model of Hormozdiari *et al.*[45] fits much better as the spread of F_c^h overlaps with that of F_o, and most of F_c^h's spread is negative just as F_o. On the other hand, F_c^s is entirely positive and has little overlap with F_o. The KS statistic tells the same story, with 0.4373 for Hormozdiari *et al.*[45] and 0.9782 for Schweiger *et al.*[46].

In, Fig. 6, we plot the distribution of the same statistics from full network realizations drawn from the two models, as well as the point estimate from the full PPI network. We use L_c^h and L_c^s as the full network analogs to F_c^h and F_c^s, respectively, and S_o to denote the point estimate for the full PPI network. For clustering coefficient, L_c^h and L_c^s look very similar, so this comparison would not lead to a different conclusion. For triangle count, L_c^s visually appears somewhat closer to S_o than L_c^h. The spread of L_c^s also contains S_o, albeit barely. However, L_c^s is also much more variable than L_c^h. In fact, L_c^s's spread reaches farther than that of L_c^h on both ends. Based on L_c^h, L_c^s and S_o, it is not obvious which model fits better, whereas our method gives a clear numerical ordering between the two models. For degree assortativity, the entirety of L_c^h is closer to S_o than L_c^s, so this comparison would not lead to a different conclusion just as clustering coefficient. Finally, since our method provides a joint distribution of the three statistics from each model as well as the PPI network, we are able to quantify overall fit that takes all three statistics into account jointly via a distance between the joint distributions (such as the multidimensional KS statistic as discussed earlier). This example demonstrates that considering the full resampling distributions, rather than point estimates as existing methods do, results in a more nuanced comparison of network models with empirical data.

Additionally, in Fig. 5, we plot the subsamples from two individual networks drawn from each model against the subsamples from independent networks drawn from each model. For each statistic, the spread and location of the two types of subsamples are similar, although triangle count shows a little more deviation than the other two since it is a sum rather than a mean. This is likely due to the rather large seeds (50 and 40 nodes) both models use as well as the rather small portion of nodes in each subsample (~30%), reflecting our observations in earlier sections.

Discussion

Network models are able to model increasingly complex dependencies that arise in network data. Yet this very dependency poses a statistical challenge, especially in the case of a single observed network. We propose a bootstrap subsampling procedure as a basis for statistical procedures in this setting that is based on a flexible resampling distribution built from the single observed network and demonstrate the procedure in both simulation and empirical test settings.

Given any network statistic of interest, its resampling distribution from the observed network can be compared against its analog from a null/candidate model based on any attribute of the distributions, such as location, spread, shape, measures of mean, and through pairwise distances. In comparison, existing methods in this setting typically rely on the point estimate from the observed network, which leads to a more limited comparison. As seen in our empirical example, this additional layer of information can sometimes lead to a different conclusion than existing methods. In addition, the distance between the resampling distributions serves as an overall measure for comparison and provides an ordering of different network models.

The flexibility of our approach is not limited to what one can do with the resampling distributions, but also extends to the type of subsampling used to generate them. Although here we used simple random samples of the nodes of the network, other schemes are possible. In fact, any method of subsampling is valid as long as it is applied to both the observed and model generated data. Thus, it can be tailored to the needs of the investigator, including statistical or computational considerations. The method of subsampling can be also used as a sensitivity analysis to see whether the results of the analysis remain unchanged under different methods of subsampling. This consideration for different methods of subsampling motivates the most immediate step for future work as it begs the question whether they can lead to performance gains. Perhaps certain types of subsampling schemes can outperform others given the method of sampling used to obtain the observed data.

Received: 10 June 2019; Accepted: 25 October 2019;
Published online: 13 November 2019

References

1. Newman, M. *Networks: an introduction* (2010).
2. Wasserman, S. & Faust, K. *Social network analysis: Methods and applications*, vol. 8 (Cambridge university press, 1994).
3. Pastor-Satorras, R. & Vespignani, A. *Evolution and structure of the Internet: A statistical physics approach* (Cambridge University Press, 2007).
4. Lusher, D., Koskinen, J. & Robins, G. *Exponential random graph models for social networks: Theory, methods, and applications* (Cambridge University Press, 2013).
5. Raval, A. & Ray, A. *Introduction to biological networks* (CRC Press, 2013).
6. Watts, D. J. *Six degrees: The science of a connected age* (WW Norton & Company, 2004).
7. Robins, G., Pattison, P., Kalish, Y. & Lusher, D. An introduction to exponential random graph (p*) models for social networks. *Soc. networks* **29**, 173–191 (2007).
8. Hoff, P. D., Raftery, A. E. & Handcock, M. S. Latent space approaches to social network analysis. *J. american Stat. association* **97**, 1090–1098 (2002).
9. Goyal, R., Blitzstein, J. & De Gruttola, V. Sampling networks from their posterior predictive distribution. *Netw. Sci.* **2**, 107–131 (2014).
10. An, W. Fitting ergms on big networks. *Soc. science research* **59**, 107–119 (2016).
11. Besag, J. Spatial interaction and the statistical analysis of lattice systems. *J. Royal Stat. Soc. Ser. B (Methodological)* 192–236 (1974).
12. Geyer, C. J. & Thompson, E. A. Constrained monte carlo maximum likelihood for dependent data. *J. Royal Stat. Soc. Ser. B (Methodological)* 657–699 (1992).
13. Snijders, T. A. Markov chain monte carlo estimation of exponential random graph models. *J. Soc. Struct.* **3**, 1–40 (2002).
14. Van Duijn, M. A., Gile, K. J. & Handcock, M. S. A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models. *Soc. Networks* **31**, 52–62 (2009).
15. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *science* **286**, 509–512 (1999).
16. Watts, D. J. & Strogatz, S. H. Collective dynamics of 'small-world' networks. *nature* **393**, 440–442 (1998).
17. Solé, R. V., Pastor-Satorras, R., Smith, E. & Kepler, T. B. A model of large-scale proteome evolution. *Adv. Complex Syst.* **5**, 43–54 (2002).
18. Vázquez, A., Flammini, A., Maritan, A. & Vespignani, A. Modeling of protein interaction networks. *Complexus* **1**, 38–44 (2003).
19. Klemm, K. & Eguiluz, V. M. Highly clustered scale-free networks. *Phys. Rev. E* **65**, 036123 (2002).
20. Kumpula, J. M., Onnela, J.-P., Saramäki, J., Kaski, K. & Kertész, J. Emergence of communities in weighted networks. *Phys. review letters* **99**, 228701 (2007).
21. Efron, B. Nonparametric estimates of standard error: the jackknife, the bootstrap and other methods. *Biom.* **68**, 589–599 (1981).
22. Good, P. I. *Resampling methods* (Springer, 2006).
23. Wu, C.-F. J. Jackknife, bootstrap and other resampling methods in regression analysis. *Annals Stat.* 1261–1295 (1986).
24. Hunter, D. R., Goodreau, S. M. & Handcock, M. S. Goodness of fit of social network models. *J. Am. Stat. Assoc.* **103**, 248–258 (2008).
25. Shore, J. & Lubin, B. Spectral goodness of fit for network models. *Soc. Networks* **43**, 16–27 (2015).
26. Desmarais, B. A. & Cranmer, S. J. Statistical mechanics of networks: Estimation and uncertainty. *Phys. A: Stat. Mech. its Appl.* **391**, 1865–1876 (2012).
27. Ohara, K., Saito, K., Kimura, M. & Motoda, H. Resampling-based framework for estimating node centrality of large social network. In *International Conference on Discovery Science*, 228–239 (Springer, 2014).
28. Bhattacharyya, S. *et al*. Subsampling bootstrap of count features of networks. *The Annals Stat.* **43**, 2384–2411 (2015).
29. Ali, W., Wegner, A. E., Gaunt, R. E., Deane, C. M. & Reinert, G. Comparison of large networks with sub-sampling strategies. *Sci. reports* **6**, 28955 (2016).
30. Thompson, M. E., Ramirez Ramirez, L. L., Lyubchich, V. & Gel, Y. R. Using the bootstrap for statistical inference on random graphs. *Can. J. Stat.* **44**, 3–24 (2016).
31. Gel, Y. R., Lyubchich, V. & Ramirez, L. L. R. Bootstrap quantification of estimation uncertainties in network degree distributions. *Sci. reports* **7**, 5807 (2017).
32. Stumpf, M. P., Wiuf, C. & May, R. M. Subnets of scale-free networks are not scale-free: sampling properties of networks. *Proc. Natl. Acad. Sci.* **102**, 4221–4224 (2005).
33. Wood, C. L. & Altavela, M. M. Large-sample results for kolmogorov-smirnov statistics for discrete distributions. *Biom.* **65**, 235–239 (1978).
34. Cooper, C. & Frieze, A. A general model of web graphs. *Random Struct. & Algorithms* **22**, 311–335 (2003).
35. Li, S., Choi, K. P. & Wu, T. Degree distribution of large networks generated by the partial duplication model. *Theor. Comput. Sci.* **476**, 94–108 (2013).
36. Erdős, P. & Rényi, A. On random graphs i. *Publ. Math. Debrecen* **6**, 290–297 (1959).
37. Pastor-Satorras, R., Smith, E. & Solé, R. V. Evolving protein interaction networks through gene duplication. *J. Theor. biology* **222**, 199–210 (2003).
38. Polley, E. C., Rose, S. & Van der Laan, M. J. Super learning. In *Targeted Learning*, 43–66 (Springer, 2011).

39. Van der Laan, M. J., Polley, E. C. & Hubbard, A. E. Super learner. *Stat. applications genetics molecular biology* 6 (2007).
40. Chen, S., Mira, A. & Onnela, J.-P. Flexible model selection for mechanistic network models. *J. Complex Networks* (2019).
41. Peacock, J. Two-dimensional goodness-of-fit testing in astronomy. *Mon. Notices Royal Astron. Soc.* **202**, 615–627 (1983).
42. Fasano, G. & Franceschini, A. A multidimensional version of the kolmogorov–smirnov test. *Mon. Notices Royal Astron. Soc.* **225**, 155–170 (1987).
43. Justel, A., Peña, D. & Zamar, R. A multivariate kolmogorov-smirnov test of goodness of fit. *Stat. & Probab. Lett.* **35**, 251–259 (1997).
44. Salwinski, L. *et al*. The database of interacting proteins: 2004 update. *Nucleic acids research* **32**, D449–D451 (2004).
45. Hormozdiari, F., Berenbrink, P., Pržulj, N. & Sahinalp, S. C. Not all scale-free networks are born equal: the role of the seed graph in ppi network evolution. *PLoS computational biology* **3**, e118 (2007).
46. Schweiger, R., Linial, M. & Linial, N. Generative probabilistic models for protein-protein interaction networks–the biclique perspective. *Bioinforma.* **27**, i142–i148 (2011).
47. Newman, M. E. Assortative mixing in networks. *Phys. review letters* **89**, 208701 (2002).

Acknowledgements

S.C. is supported by NIH U01HG009088 and U54GM088558; J.P.O. is supported by NIH R37AI051164, R01AI112339, U54GM088558, and R01AI138901.

Author contributions

S.C. conceived the method, prepared the initial manuscript, conducted the simulation and data analysis under the supervision of J.P.O. All authors reviewed and edited the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41598-019-53166-6.

Correspondence and requests for materials should be addressed to J.-P.O.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.