

Exponential Random Graph models for Little Networks*

George G. Vega Yon^{†1}, Andrew Slaughter², and Kayla de la Haye¹

¹University of Southern California, Department of Preventive Medicine

²U.S. Army Research Institute for the Behavioral and Social Sciences

July 2019

1 Introduction

Statistical models for social networks have enabled researchers to study complex social phenomena that give rise to observed patterns of relationships among social actors, and to gain a rich understanding of the *interdependent* nature of social ties and social actors [15, 11]. For example, this research has provided new insights into the role that the attributes of social actors (e.g., their characteristics, beliefs, and decisions), and endogenous structural processes (e.g., social balance, and relationship reciprocity) play in shaping social networks across different populations and social settings, and how these social networks, in turn, influence individuals and groups.

Much of this research has focused on social networks within medium to large social groups: networks ranging from dozens or hundreds of members (e.g., classrooms and organizations) to millions (e.g., online social networks). However, modern advances in statistical models for social networks have rarely been applied to the study of small networks, despite small network data from teams, families, and personal (ego-centric) networks being common in many fields that study social phenomena [?, ?, ?, ?]. The study of small networks often uses descriptive statistics that summarize basic structural features of the network; for example, the density, degree distribution, or triad count. However, researchers in these fields are often

*The authors would like to thank Garry Robins, Carter Butts, Johan Koskinen, Noshir Contractor, and Andrew Slaughter for their valuable contributions to this work. The usual disclaimer applies. © 2019. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

[†]Corresponding author. email: vegayon@usc.edu

interested in testing hypotheses about *why* localized social structures, such as reciprocity, balance, and homophily, emerge in these small groups. A key limitation to such work has been the availability of statistical models for networks that can flexibly test and control for the kind of dependencies inherent to network data. In this paper, we propose an approach for applying one of the most widely used statistical models for social networks—exponential random graph models, or ERGMs—to small graphs, to better enable new and rich research on “little networks”.

2 Exponential-Family Random Graph Models

Exponential-family random graph models (ERGMs) are one of the most popular tools used by social scientists to understand social networks and test hypotheses about these networks [13, 7, 3, 21, 16, and others]. In this family of models, an observed graph y , comprised of a set of nodes (vertices) and ties (edges), is characterized by a set of sufficient statistics defined on the graph, $s(y)$, and parameters θ . In a model that also includes node characteristics X , this leads to the following equation:

$$\Pr(Y = y \mid \theta, X) = \frac{\exp\{\theta^t s(y, X)\}}{\kappa(\theta, X)}, \quad \forall y \in \mathcal{Y} \quad (1)$$

Where $\kappa(\theta, X) = \sum_{y \in \mathcal{Y}} \exp\{\theta^t s(y, X)\}$ is the normalizing constant, and \mathcal{Y} is the support of the model which is usually assumed to include all graphs of the same type (e.g., directed or undirected) and size, and do not include self-ties. In the directed graph case, the size of \mathcal{Y} equals $2^{n(n-1)}$ possible graphs. This makes the exact calculation of $\kappa(\theta, X)$, and therefore of (1), computationally expensive. A sophisticated array of parameters can be specified for ERGMs that reflect social and structural process of interest to social scientists, such as social closure, connectivity, and other affiliation preferences. Figure 1 shows some examples of the structures (statistics) that can be estimated with ERGMs.

Although “small networks” is a topic mentioned several times in the literature on social network models [21, 3, 15], interest in larger social networks has dominated the field.¹ Thus, ERGM methods have been developed to accommodate larger networks (although they do not scale well to “very large” networks of several thousand nodes or more). One example of this is the calculation of the likelihood function: rather than being calculated using exhaustive enumeration (which we will refer to as “exact likelihood”), the most popular software packages used for estimating these models apply simulation-based estimation methods. As a consequence, current methods used to estimate ERGMs for medium to large networks do

¹This is perhaps because, as put by [15], small networks are considered to be “uninteresting special cases”

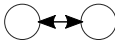
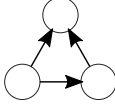

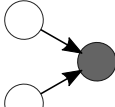
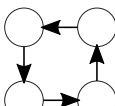
Representation	Description
	Mutual Ties (Reciprocity) $\sum_{i \neq j} y_{ij} y_{ji}$
	Transitive Triad (Balance) $\sum_{i \neq j \neq k} y_{ij} y_{jk} y_{ik}$
	Homophily $\sum_{i \neq j} y_{ij} \mathbf{1}(x_i = x_j)$
	Covariate Effect for Incoming Ties $\sum_{i \neq j} y_{ij} x_j$
	Four Cycle $\sum_{i \neq j \neq k \neq l} y_{ij} y_{jk} y_{kl} y_{li}$

Figure 1: Besides of the common edge count statistic (number of ties in a graph), ERGMs allow measuring other more complex structures that can be captured as sufficient statistics.

not translate well to small network data (i.e., 10 or fewer nodes), and applications of these statistical network models to small networks are rare.

One major technical and theoretical issue in ERGM estimation generally, which is exacerbated with small networks, is the problem of *degeneracy*. Degenerate models occur when the observed graph statistics lie in a region on or near the boundary of the support, and can be stressed when estimation depends on Monte Carlo Integration [17, 5]. Small networks, which are more likely to be nearly empty or nearly full, have a smaller region of support, and are more likely to be on or near the boundary of that support. For example, if we are trying to estimate an ERGM in a network with only three nodes, in the scenario where the graph is directed and does not allow for self-ties, the chances of obtaining a graph with either one or zero ties (i.e., empty or almost completely empty), or a graph with five or six ties (i.e., fully or almost fully connected) is about 20% using a uniform sampler. The degeneracy problem is so common with small networks (e.g., three to ten nodes).

Because researchers studying small networks often have observed *samples* of small networks (e.g., multiple team, family, or personal/egocentric networks), a common work-around to the issue of model degeneracy is to combine the independent small networks into a single larger block-diagonal graph. Estimation then proceeds by assuming that ties between blocks are impossible (i.e., treated as structural zeros in estimation). The major problems with this approach are that it can be complicated to fit, and difficult to extend. For example, the same set of constraints (the structural zeros) that allows for the model to be fit can also make

the estimation procedure more difficult, and increase the possibility of sampling problems during MCMC estimation. More importantly, however, the block-diagonal approach can be difficult to extend. For example, a basic "complete pooling" model that assumes a common data generating process across all networks is straightforward to define using existing tools for ERGM estimation. However, relaxing that assumption to allow for variability across graphs (i.e., an unpooled model) can become problematic; this would typically require the creation of block-wise node membership attributes, and complex interaction terms involving subgraph statistics and node membership variables. Moreover, extending such a framework to not just allow for between-group variability, but to *predict* it (for example, as a function of additional group-level variables), is not straightforward in such an approach.

To overcome the challenges described above for fitting ERGMs to small networks, we leverage the fact that in the case of small networks, the full likelihood function *is* tractable. This allows the direct estimation of model parameters without using Markov Chain Monte Carlo (MCMC) or other approximate methods, avoiding some of the convergence issues associated with inference degeneracy [5]. It also opens up the possibility of making it much easier to combine ERGMs with other statistical techniques, opening the door for richer methods for modeling and understanding small-group structure and dynamics. In this paper, we describe how modern computational power allows for the complete specification of the likelihood for small graphs, and how this specification allows us to use the standard tools of maximum likelihood estimation (MLE), as opposed to approximate methods, including confidence intervals and likelihood ratio tests. We present examples using these techniques, provide some initial Monte Carlo results on bias, type I error rates, and power, and discuss future extensions these techniques make feasible.

3 ERGMitos: ERGMs for small networks

With modern computers, calculating the exact likelihood function of an ERGM for a small network becomes computationally feasible. This has an important implication: the process for estimating the parameters of an ERGM for small networks can be done directly. Many innovative techniques have been developed to handle models with intractable normalizing constants (e.g., MCMC-MLE, Bayesian techniques such as the exchange sampler, etc.), and often these techniques work quite well. Of course, no techniques are without tradeoffs; MCMC-MLE estimation can be very sensitive to starting values, and the quality of standard errors can depend on the availability of analytic gradients [?]. Bayesian techniques like the exchange sampler [?] may be comparatively slow, which may be an issue when many networks are to be analyzed.

Moreover, simulation-based methods may have particular susceptibilities to model degeneracy. As stated by [5, p. 7], "[i]f the model used to simulate the graphs is not close enough to produce realizations that cover the observed values of the statistics, the MC-MLE will not exist even in cases where the MLE does." For example, many common network models, such as triangle-based models, can lead to bimodal distributions of graph statistics that simulation-based methods have difficulty with, even when the MLE falls between the modes [?]. Therefore, even though the degeneracy issue is not completely avoided, a method based on exact (non-simulation) inference may not only provide a better solution (in general) by avoiding the additional uncertainty induced by simulations and approximations, but may help mitigate the degeneracy issue in cases where the MLE exists.

Of course, the statistical analysis of a single small network is going to be uninformative due to the small numbers of dyads, and a high restriction in the variability of possible subgraph statistics. Fortunately, small network data is typically collected from *samples* of small groups, which allows for the development of models to analyze structural variation both within and across small networks. If we assume that the sample of networks comes from a population of networks (groups) that are governed by the same data generating process, we end up with the following likelihood, defining a completely-pooled model:

$$\Pr(Y_1 = y_1, \dots, Y_P = y_P \mid \theta, X_1, \dots, X_P) = \prod_{p=1}^P \frac{\exp\{\theta^t s(y_p, X_p)\}}{\kappa_p(\theta, X_p)} \quad (2)$$

Where P denotes the number of networks used in the model, and $\kappa_p(\theta, X_p)$ is explicitly calculated, unlike existing approaches to ERGM estimation. We call this framework, which is a revisited version of ERGM in the case of small networks, *ERGMito*².

One complication with ERGMs involving multiple networks is that such models are usually not projective [?], and the magnitude of parameter estimates and their standard errors (at least with network statistics in standard use) are all implicitly conditioned on network size. This means caution is required when combining (or comparing) parameter estimates from different networks. Even the interpretation of the basic edge count parameter can become difficult; as pointed out in [9], assuming a common edge parameter in networks of different sizes is equivalent to assuming equal edge probabilities (verify this doesn't need qualification, e.g., conditional probs). This implies that one assumes network density increases as network size increases, but many real-world social networks are sparse and density grows much more slowly than network size.

This is a significant concern in cases where samples of networks may span a wide range of

²The *ito/ita* suffix is used in Spanish to denote small, or affection. We are especially grateful to George Barnett who proposed the name during the North American Social Networks Conference in 2018.

sizes, and requires careful consideration of issues regarding choice of network statistics and the appropriateness of a completely-pooled approach (compared to, for example, a partially-pooled approach that allows for variability in estimates across networks). However, in the case outlined in this paper we focus on a sample of networks that are of similar size, and we assume there is a common data-generating process. The requirements and assumptions of the completely-pooled model in (2) therefore can be assumed to hold.

The simulations and model fitting were conducted using the R package `ergmito`, which has been developed to implement the methods described in this paper.

4 Illustration with simulated data: fivenets

4.1 Data-generating-process and model fitting

In the following section we work with a simulated data set that was created using the data-generating-process of `ERGMitos`. This particular dataset, which we call “fivenets”, is included in the in the R package `ergmito`³. The data set contains five simulated networks with nodal attributes (we use gender in the following example), and the networks were generated using the following specification:

$$\Pr(Y = y \mid X_{\text{gender}}, \theta) = \frac{\exp \left\{ \theta_{\text{edges}} \left(\sum_{i,j} y_{ij} \right) + \theta_{\text{homophily}} \left(\sum_{i,j} y_{ij} \mathbf{1}(X_{\text{gender},i} = X_{\text{gender},j}) \right) \right\}}{\kappa(\theta, X)}$$

where $\theta_{\text{edges}} = -2.0$ and $\theta_{\text{homophily}} = 2$. Using this equation we draw five networks of size five. The process of “homophily” is represented by a parameter that is defined as the number of ties in which ego and alter have the same gender. Before drawing the networks we randomly generated the node attribute (gender) parameter to each vertex as a Bernoulli with parameter 0.5. [Figure 2](#) shows the generated networks, including their nodal attributes.

Using the `ergmito` R package, we fit three different models to the data⁴: (1) a Bernoulli graph, which is a model that only includes the “edges” parameter, (2) a model with “gender homophily” as its only parameter, and finally (3) a model including both “edges” and “gender homophily”, this is, the correct specification of the model. [Table 1](#) shows the estimation results of the three different specifications of the model and as expected, model (3) has the

³The R package is available to be downloaded at <https://github.com/muriteams/ergmito>.

⁴Some details regarding the computational aspects of the model fitting process are provided in [A](#).

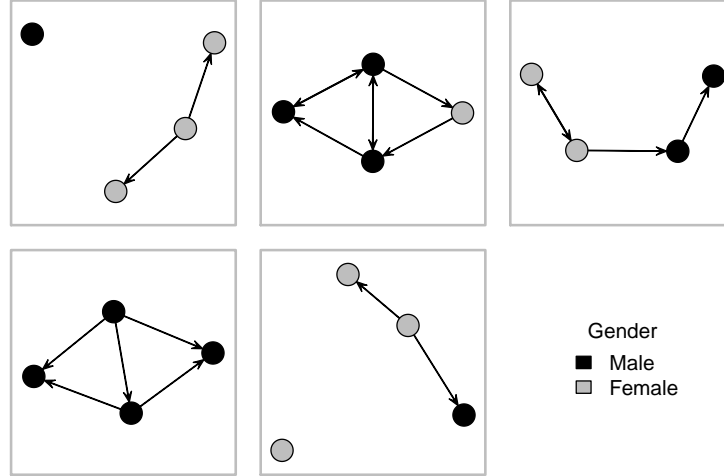


Figure 2: Fivenets data set. These graphs were randomly drawn from an ERGM distribution with two parameters: number of edges and gender homophily, with parameters equal to -2.0 and 2.0 respectively.

best overall fit to the data. Furthermore, since all three models were fitted using MLE, we can compare the edgcount and homophily models with the full model using Likelihood Ratio tests [23]. The *goodness of fit* of this model is evaluated in the following section.

4.2 Goodness-of-fit in ERGMitos

Researchers that apply ERGMs should be familiar with the goodness-of-fit (GOF) diagnostics that are used to assess how well the estimated model can reproduce graphs that are similar to the observed graph on a range of local and global graph statistics [?]. In the case of ERGMitos applied to small networks, local graph statistics will be more relevant than global statistics to assess GOF. For example, the graph geodesic distribution (i.e., the distribution of shortest-path lengths) is often used to assess GOF for larger networks, but this is clearly less relevant in the case of small networks (less than 10 nodes) because the shortest-path length between any two nodes typically lies between 1 and 2 steps. Therefore, we focus the GOF analysis on the parameters fit in the model as the bare-minimum of local graph statistics, as shown in Figure 3. An important difference in our approach compared to traditional GOF assessments for ERGMs is that we are able to enumerate the full support of the model, and so instead of showing a boxplot we present a 90% exact confidence interval per-statistic per network, comparing the fitted model’s distribution with the observed parameters. A detailed discussion of this aspect of the ERGMitos is presented at the end of this paper.

An important advantage of the ERGMitos over “regular” ERGMs is that we can observe the surface of the log-likelihood over different combinations of parameters in a rather

	Homopholy	Edgecount	Full model
Edgecount		−0.69*	−1.70**
		(0.27)	(0.54)
Homophily (on Gender)	−0.12		1.59*
	(0.34)		(0.64)
LR-test statistic (χ^2)	7.04**	13.72***	
AIC	85.06	78.38	73.34
BIC	87.15	80.48	77.53
Log Likelihood	−41.53	−38.19	−34.67
Num. networks	5	5	5
Convergence	0	0	0

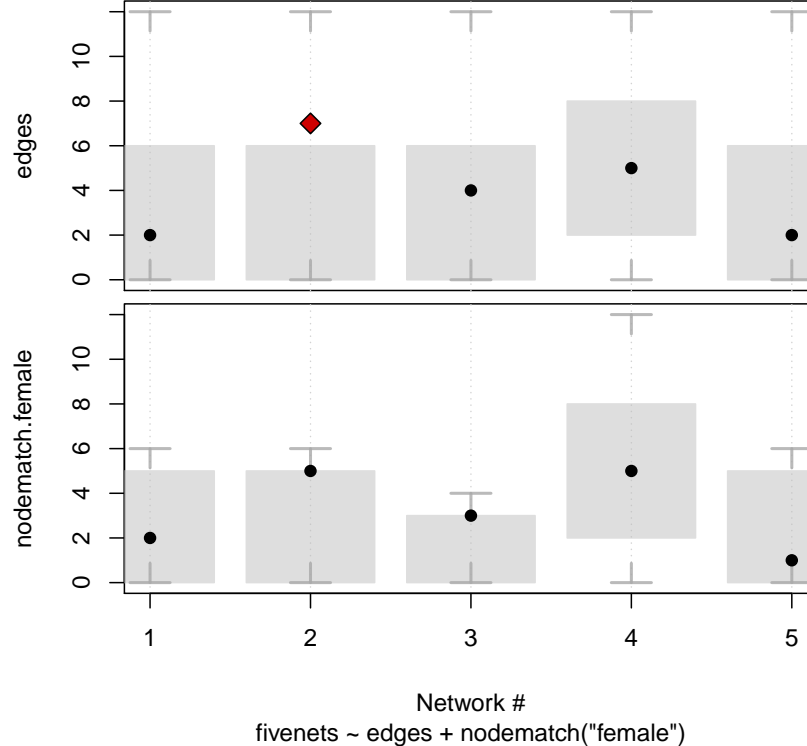
*** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$

Table 1: Fitted ERGMitos using the fivenets dataset. Looking at AICs and LR-test statistics, the full model (last column of the table) is the one with the best fit to the observed data. More over, the 95% level CI of each covers the true parameters: $\hat{\theta}_{edges} \in [-2.77, -0.64]$; $\hat{\theta}_{Homophily} \in [0.33, 2.85]$.

straightforward way. This, together with the GOF analysis should be a routine step done after every ERGMito fit. Figure 4 shows the surface of the log-likelihood function around the solution parameters to the maximization problem.

The ability to calculate the surface of the exact likelihood function, allows us to assess the quality of the estimated set of parameters. One good use of this diagnostic is to evaluate the roughness of the log-likelihood function, which in principle should give us an idea of the likelihood of the maximization process failing to reach a global maxima.

Figure 3: Goodness-of-fit in ERGMito. This illustrates how the observed sufficient statistics of each one of the 5 networks (x-axis) locate in the overall estimated distribution based on the fitted ERGMito. The gray lines in each box show the minimum and maximum value that the sufficient statistics can take in each one of the 5 networks, whereas the dotted lines provide a 90% confidence interval. The dots are the observed statistics in each network.

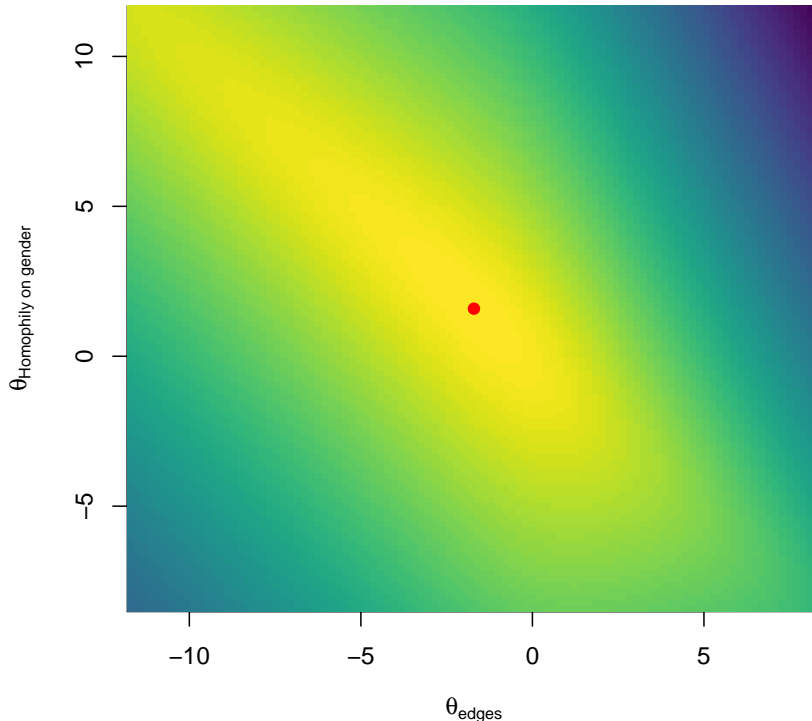


5 Simulation study

We conducted two sets of simulations studies in which we compare the performance of MLE with MC-MLE in terms of bias, power, and type I error rates. In the first set, we analyze empirical bias and empirical power of each estimator in an scenario in which the ERGM is defined by *edgecounts* and *transitive triads*. For the second set of simulations, we look at empirical type I error rates when ERGMs are misspecified by including a *transitive triad* term in the context of a data-generating-process that only includes an *edgecount* statistic.

The code used to reproduce this entire section can be found at <https://github.com/muriteams/ergmito-simulations>.

Figure 4: Surface of the log-likelihood function of the pooled ERGM*ito* model. Lighter colors represent higher values while darker ones represent lower values. The red dot corresponds to the location of the MLE estimate of the model.



5.1 Empirical Bias and Power

Using the ERGM*ito* R package, we generated 20,000 samples (datasets) consisting of several small networks. Each sample was generated using different combinations of parameters. While all come from an ERGM model defined by edgcounts and number of transitive triads, for every sample we specified: (1) population parameters for the ERGM, (2) size of the sample (number of networks in the sample), and (3) composition of the sample in terms of the combination of networks of size four and five. A detailed description of each one of these three components used to draw the samples follows:

1. **Population parameters** First we drew two numbers from a piece-wise Uniform distribution with values in $[-2, -.1] \cup [.1, 2]$, $(\theta_{edges}, \theta_{ttriads})$, which corresponded to the parameters associated to the statistics **edgcount** and **number of transitive triads**. This specifies the ERGM from which we will be drawing the networks from. This is akin to [14] approach to simulate ERGM from outside of degenerate areas, although

we took a more conservative approach than their ranges of $(-5, 0)$ and $(0, 5)$ for the parameters “edges” and “triangles” in order to increase the number of non-degenerate samples, this is, samples composed mostly by either empty or fully connected graphs.

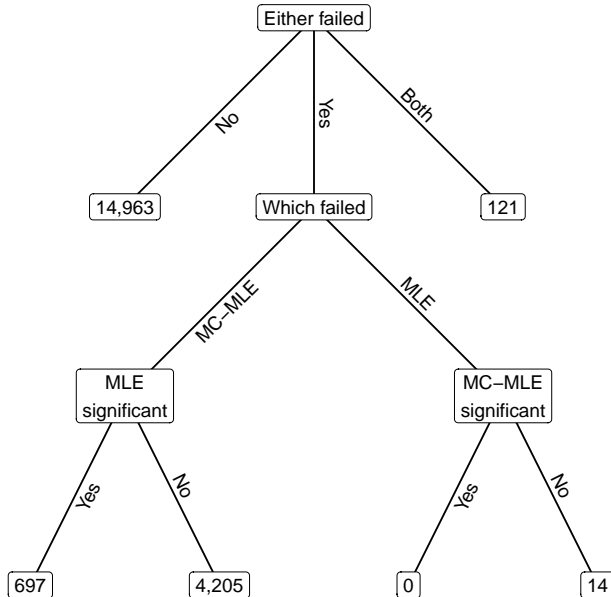
2. **Number of networks per sample** Then, we specified the number of networks to generate from the model defined in the previous step, using one of the following sample sizes $\{5, 10, 30, 50, 100, 150, 200, 300\}$. Notice that overall, for each one of the possible set of sample sizes, we ended up generating 2,500 samples for each sample size, in other words, we generated 2,500 samples with 5 networks, 2,500 samples with 10 networks, etc.
3. **Number of nodes per network** Finally, the composition of each sample in terms of the number of vertices that each network has was uniformly-random selected from the pairs $\{N, 0\}, \{N - 1, 1\}, \dots, \{1, N - 1\}, \{0, N\}$, where the first number of each pair is the number of networks of size 4, and the second is the number of networks of size 5 in the sample, for example, if the sample size selected in the previous step was 30, then the possible pairs to select from would be $\{30, 0\}, \{29, 1\}, \dots, \{1, 29\}, \{0, 30\}$, so samples in which all networks were of size 4 (meaning we draw the pair $\{30, 0\}$) or size 5 (again, selecting the pair $\{0, 30\}$) were equally likely.

For each one of the 20,000 simulated datasets, we then estimated the model using both MLE—as implemented in the *ERGMito* R package [18]—and MC-MLE—using *statnet*’s *ergm* R package [6, 8]. In the case of the latter, the pooled estimation was done by fitting what is known in the literature as a block-diagonal model in which (a) networks are stacked together in a single adjacency matrix, and (b) the sampling space for the MCMC process is constrained to sample from graphs where ties are only possible within blocks. We set the MCMC control parameters *interval* and *samplesize* to 2,048 (double the default values) to increase precision of the parameter estimates.

5.1.1 Empirical Bias and Power: Samples in which either of MC-MLE or MLE returned with an error

Of the models fit to the 20,000 datasets (each comprised of samples of small networks), in 14,963 cases the models were successfully fit using both the MLE and MC-MLE estimators. In 121 cases both implementations failed with an error, and for the remaining cases (4916) the MC-MLE (*ERGM*) failed in most of them (4905 of 4916), while implementation of the MLE (*ERGMito*) estimator failed in only 14 of these 4916 cases. Of the 4905 cases in which the MC-MLE failed, MLE was able to detect a significant effect in both parameters in 697

Figure 5: Distribution of the failed estimations. A failed estimation attempt refers to the case in which the implementation of the algorithm (ERGM for the MC-MLE, and ERGMito for the MLE method), returned with an error.



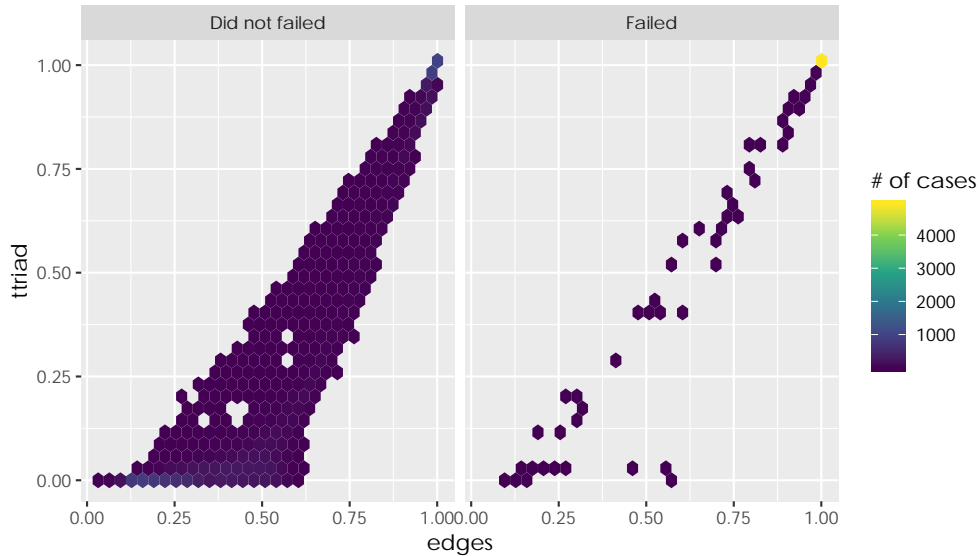
cases (i.e., 14 percent of these cases). Conversely, of the 14 cases in which the MLE failed, MC-MLE was not able detect a significant effect in any of these cases.. [Figure 5](#) summarizes the succesful and failed models described above.

To gain insight into the problems causing the failed models, we considered the distribution of the sufficient statistics. These indicate that most of the problems arose in situations where the sufficient statistics were near to the boundary of the support. This is illustrated in [Figure 6](#), and shows that in most of the cases in which both methods failed, the sufficient statistics were mostly saturated, meaning they reached the maximum possible value because the cases sampled were largely composed of fully connected networks. Given errors arose when the set of sufficient statistics was close to saturation, both ERGM and ERGMito should consider those cases and test for possible failures during the estimation process.

We also considered if the rate of model failure was greater when the dataset contained a smaller sized sample of networks. [Table 2](#) shows the error rates by sample size for both implementation methods, and it is clearly higher for smaller samples (particularly less than 30 networks), and this is especially true for MC-MLE.

Moreover, if the optimization algorithms are able to return anything, both ERGM and ERGMito try to compute variance-covariance matrices using the model’s log-likelihood; this being approximated in the case of MC-MLE. It is important to note that in the case that the Hessian matrix is not p.s.d (positive-semi-definite), both ERGM and ERGMito use the

Figure 6: Distribution of the average sufficient statistics per sample. Since samples can contain networks of sizes four and five, we have re-scaled the sufficient statistics counts by each network size’s corresponding maximum value so these range from zero to one.



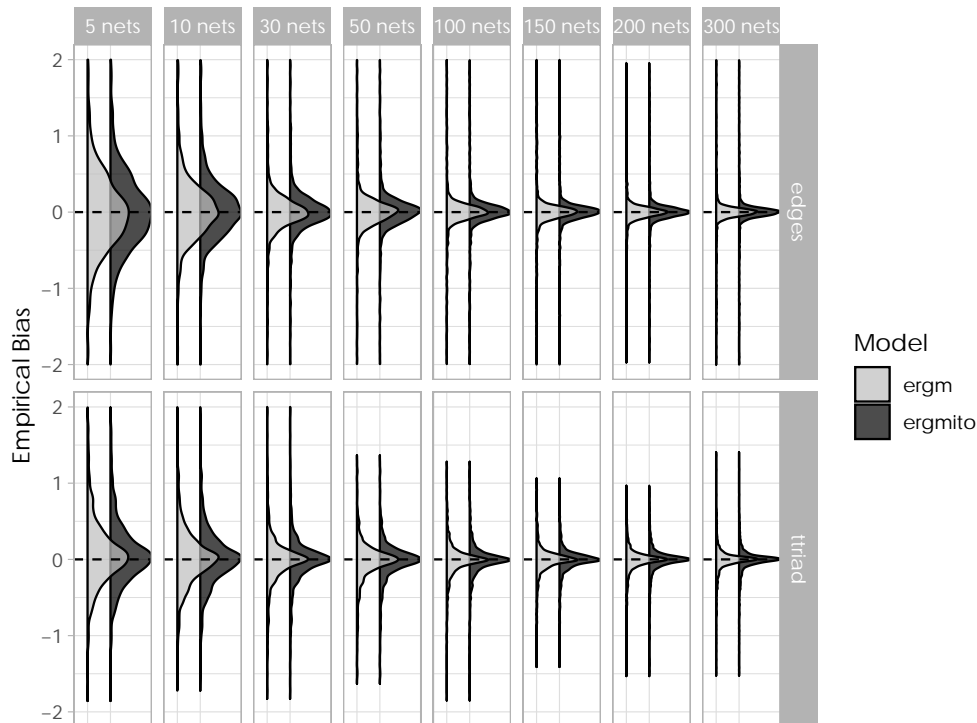
Moore-Penrose generalized inverse algorithm as implemented in the R package MASS [20]. For more on the interpretation of variance-covariance matrices when the Hessian is not p.s.d., see [4]. Details on the evaluation of parameter estimates in ERGMIto can be found in [Appendix B](#).

Sample size	P(Error)	
	MC-MLE	MLE
5	0.35	0.00
10	0.32	0.03
30	0.27	0.01
50	0.25	0.01
100	0.23	0.00
150	0.20	0.00
200	0.20	0.00
300	0.18	0.00

Table 2: Failure probability (software implementation) by sample size.

Finally, it is important to highlight that unexpected errors during the estimation process, despite being associated with properties of each estimation method, are issues that could be resolved through software development and so can be improved.

Figure 7: Empirical distribution of the bias per model parameter, for MC-MLE and MLE estimation methods. In general we see that the parameter estimates' bias is centered around zero and both MC-MLE (ERGM) and MLE (ERGMito) have about the same bias in our simulation study.



5.1.2 Empirical Bias and Power: Samples fitted successfully

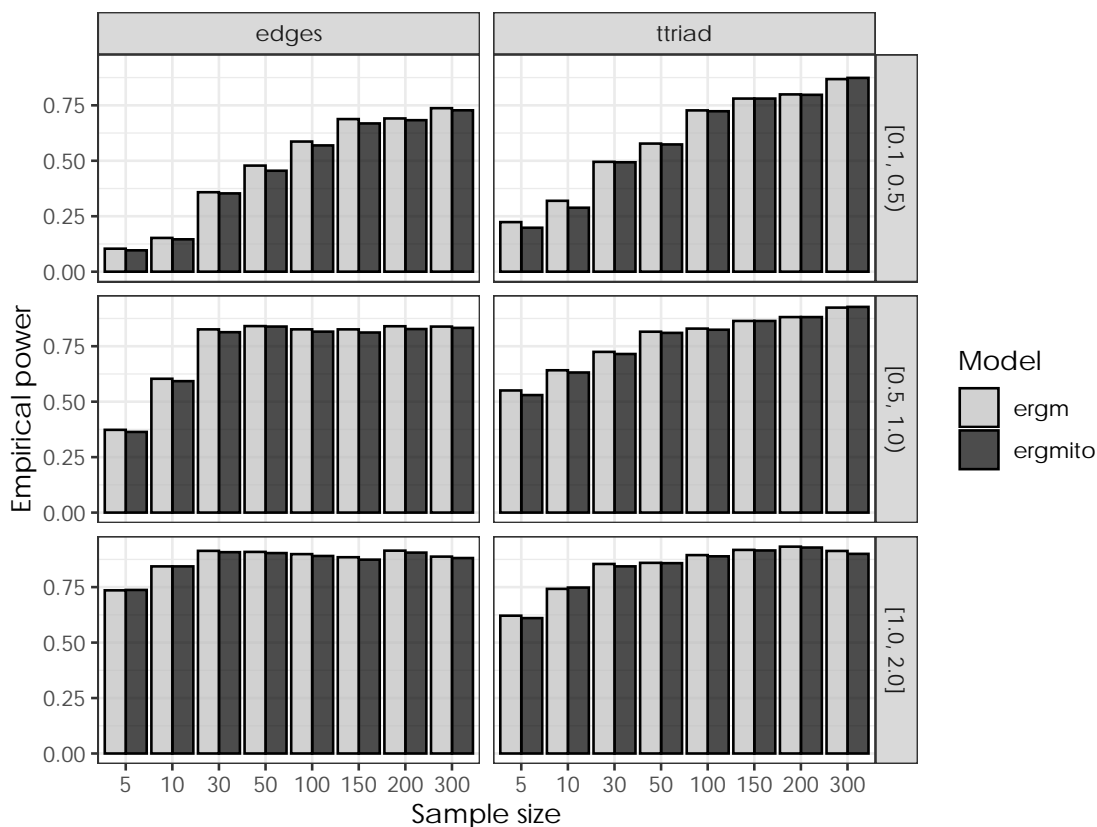
Out of the 20,000 model fitting attempts on the simulated datasets, about 25% of those fit with either MLE or MC-MLE failed to return an estimate. The results that follow only include those samples in which both MLE and MC-MLE methods returned without an error. As shown in Figure 7, both estimation methods behaved similarly in terms of empirical bias in the models studied here. As the size of the sample of networks in the dataset increased (i.e., when there were more networks within the sample), the empirical bias of both MC-MLE and MLE decreased, as expected.

As shown in Figure 8, both MC-MLE and MLE behave very similarly with empirical power levels increasing as the size of the sample of networks increases in the dataset size (the x-axis of each sub-figure), and as the effect size increases (the y-axis of each sub-figure). Moreover, we compared power levels using two-sample proportion tests and found no significant differences between the power levels of MC-MLE and MLE. It is interesting to note that, in the case of effect sizes of magnitude $[0.5, 1.0)$, the discovery rate for the `ttriad` parameter reaches nearly 0.75 for sample sizes between 30 to 50 networks, which is a rather

common sample size in the study of small networks such as teams, families, and sometimes ego-networks.

Figure 9 shows the effect of the composition of the sample in each dataset, in terms of the proportion of networks of size five (vs. size four), and the number of networks per dataset. In this case, we observe no meaningful patterns that would indicate the dataset composition is related to power.

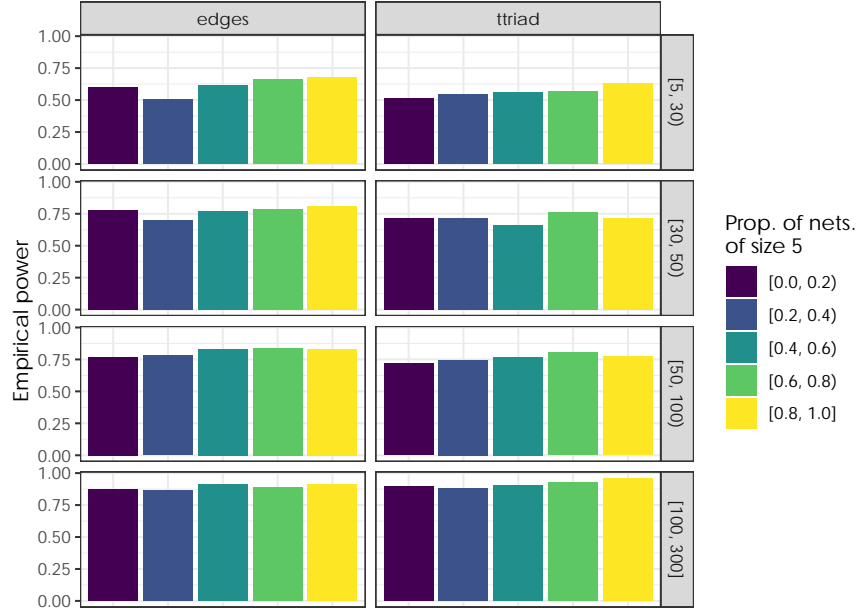
Figure 8: Empirical power by dataset size and effect size (the later considering only magnitude), for ERGM and ERGMito estimation methods. Power increases for both MC-MLE (ERGM) and MLE (ERGMito) with increases in the size of the dataset and effect size. There are indistinguishable differences in power between the two estimation methods.



One notable difference between the two estimation methods was in computational time: using MLE makes the estimation process significantly faster. As shown in Figure 10, MLE (ERGMito) is orders of magnitude faster than MC-MLE (ERGM). Therefore, while both estimators show very similar properties in terms of power and bias, practitioners will benefit by using MLE (ERGMito) when modeling small networks because it may substantially reduce the estimation time.⁵

⁵While this is mostly true, there are some scenarios in which the speed gains may not be as dramatic as it has been showed here. The biggest computational bottleneck that ERGMito faces is the calculation of

Figure 9: Empirical power by proportion of networks of size five in the sample (color coded) and sample size (rows).



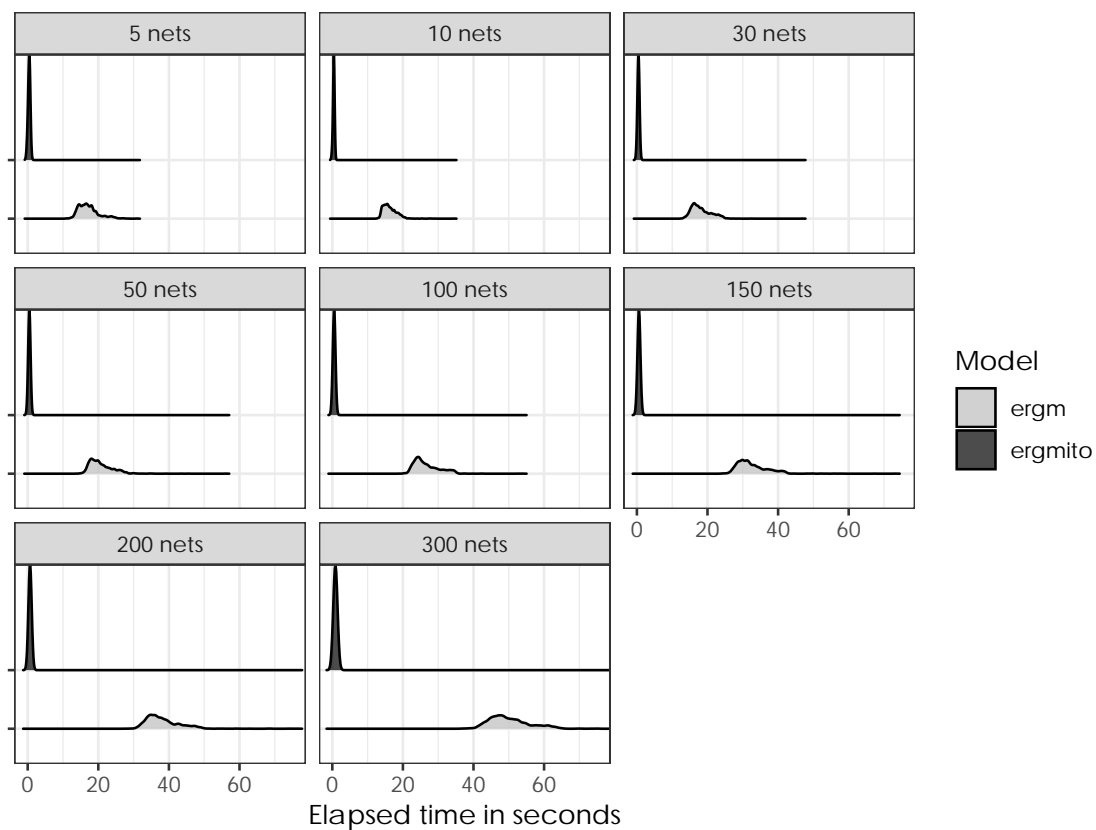
5.2 Type I error rates

Using the same procedure described in [subsection 5.1](#), we simulated 17,500 datasets comprised of Bernoulli networks (i.e., an ERGM model only defined by the *edgcounts* sufficient statistic). In this case, we drew different sets of sample sizes: for each of $\{5, 10, 15, 20, 30, 50, 100\}$ we generated 2,500 datasets using the Bernoulli model with *edgcount* parameter uniformly distributed in the range $[-2, -1] \cup [1, 2]$. We then estimated the models using MC-MLE and MLE, as implemented in the *ergm* and *ergmito* R packages respectively, and calculated the type I error rates, using a misspecified model, this is, fitting ERGMs including a *transitive triads* count statistic.

As shown in [table 3](#), when models are fit to datasets with samples sizes of 15 or less networks, the MLE estimates had a significantly smaller type I error rate at the 5% significance level, after comparing both rates using a two-sample proportion test. With datasets that had samples sizes of 20 or more networks, there was no statistically significant difference in type 1 error rates between the two methods.

the full support of the sufficient statistics. In the case of structure-only statistics, *ERGMito*, and actually *ergm*, computes the full distribution very quickly, but, as the model starts to become more complex, such calculation becomes more expensive.

Figure 10: Distribution of elapsed time (in seconds) for the estimation process for MC-MLE (ERGM) versus MLE (using *ERGMito*). Overall, the MLE implementation is orders of magnitude faster compared to the time required by the MC-MLE implementation to do the parameter estimation.



Sample size	N. Simulations	P(Type I error)		
		MC-MLE	MLE	chi2
5	2,189	0.084	0.057	11.71 ***
10	2,330	0.070	0.045	12.46 ***
15	2,395	0.084	0.066	5.55 *
20	2,430	0.074	0.060	3.58
30	2,460	0.057	0.052	0.67
50	2,495	0.046	0.044	0.17
100	2,499	0.048	0.048	0.00

Table 3: Empirical Type I error rates. The χ^2 statistic is from a 2-sample test for equality of proportions, and the significance levels are given by *** $p < 0.001$, ** $p < 0.01$, and * $p < 0.05$. The lack of fitted samples in some levels is due to failure of the estimation method.

6 Discussion

In this paper we revisit and extend Exponential Family Random Graph Models (ERGM) for the case of small networks. Given the interest in testing hypotheses about small networks in the literature, but limited application of statistical models to small network data, we proposed a special case of ERGMs for small networks, called *ERGMitos*. An appealing feature of *ERGMitos* is that they are estimated using MLE directly, rather than estimation via MCMC methods, as is traditionally done with ERGMs for larger networks. This approach provides a couple of important benefits for small network data: (1) it avoids at least partially, if not completely, the model near-degeneracy problems that affects the MC-MLE methods; and (2) its use of pooled estimates is valuable when datasets are comprised of small networks, as is often the case with research on multiple families, small teams, or ego-networks. Our simulation study finds that *ERGMitos* are easily estimated, and that this approach can improve accuracy and efficiency for modeling small networks compared to MC-MLE estimation approaches implemented in other ERGM packages.

Other topics to mention in discussion here: (1) full LL allows for simple extensions that are very difficult otherwise (use mixture model as example, perhaps mention hierarchical ERGMs since it’s relevant to the model here and the general application space of team research); (2) if derivative is available and computeable, allows for advanced Bayesian methods like HMC, MALA, etc.

The development and evaluation of *ERGMitos* in this simulation study also brings up topics for future work. One is the evaluation of model goodness-of-fit, and identifying statistics that are most important to evaluate with small networks, and that are reasonable to expect in a model that would suggest a “good fit”. Because *ERGMitos* enable a rather sim-

ple way of conducting simulation studies (relative to traditional ERGMs), this will facilitate this work in future. Another topic to explore in future work is the value of ERGMitos for estimating ERGMs for very large networks, by drawing *samples* of local network structures from a large graph. There is ongoing work extending ERGMs to very large networks [?, ?], and ERGMitos may be a valuable and *efficient* approach for fitting these pooled models to a large sample (e.g., in the order of the thousands) of small local network structures, with 5 or 6 nodes, drawn from a large network.

Overall, ERGMitos provide a promising extension to the ERGM framework for the analysis of small social networks, which can generate a richer understanding of the local social processes that give rise to the formation of networks in small social groups, and processes that are the *local social building blocks* of many larger social structures.

7 Acknowledgements

This material is based upon work support by, or in part by, the U.S. Army Research Laboratory and the U.S. Army Research Office under grant number W911NF-15-1-0577. The views, opinions, and/or findings contained in this paper are those of the authors and shall not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documents

Computation for the work described in this paper was supported by the University of Southern California’s Center for High-Performance Computing (hpcc.usc.edu).

References

- [1] Carter T. Butts. sna: Tools for Social Network Analysis, 2016. R package version 2.4.
- [2] Dirk Eddelbuettel and Romain François. Rcpp: Seamless R and C++ integration. Journal of Statistical Software, 40(8):1–18, 2011.
- [3] O Frank and David Strauss. Markov graphs. Journal of the American Statistical Association, 81(395):832–842, 1986.
- [4] Jeff Gill and Gary King. What to Do When Your Hessian is Not Invertible: Alternatives to Model Respecification in Nonlinear Estimation. Sociological Methods & Research, 33(1):54–87, 2004.
- [5] Mark S. Handcock. Assessing Degeneracy in Statistical Models of Social Networks. Working Paper No. 39, 76(39):33–50, 2003.
- [6] Mark S. Handcock, David R. Hunter, Carter T. Butts, Steven M. Goodreau, Pavel N. Krivitsky, and Martina Morris. ergm: Fit, Simulate and Diagnose Exponential-Family Models for Networks. The Statnet Project (<http://www.statnet.org>), 2018. R package version 3.9.4.
- [7] Paul W. Holland and Samuel Leinhardt. An exponential family of probability distributions for directed graphs. Journal of the American Statistical Association, 76(373):33–50, 1981.
- [8] David R. Hunter, Mark S. Handcock, Carter T. Butts, Steven M. Goodreau, and Martina Morris. ergm: A package to fit, simulate and diagnose exponential-family models for networks. Journal of Statistical Software, 24(3):1–29, 2008.
- [9] Pavel N. Krivitsky, Mark S. Handcock, and Martina Morris. Adjusting for network size and composition effects in exponential-family random graph models. Statistical Methodology, 8(4):319–339, jul 2011.
- [10] Philip Leifeld. texreg: Conversion of statistical model output in R to L^AT_EX and HTML tables. Journal of Statistical Software, 55(8):24, 2013.
- [11] D. Lusher, J. Koskinen, and G. Robins. Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications. Structural Analysis in the Social Sciences. Cambridge University Press, 2012.

- [12] R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2018.
- [13] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph (p^*) models for social networks. Social Networks, 29(2):173–191, 2007.
- [14] Michael Schweinberger and Mark S. Handcock. Local dependence in random graph models: characterization, properties and statistical inference. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 77(3):647–676, jun 2015.
- [15] Tom A. B. Snijders. Statistical Models for Social Networks. Annual Review of Sociology, 37(1):131–153, 2011.
- [16] Tom A B Snijders, Philippa E Pattison, Garry L Robins, and Mark S Handcock. New specifications for exponential random graph models. Sociological Methodology, 36(1):99–153, dec 2006.
- [17] David Strauss. On a General Class of Models for Interaction. SIAM Review, 28(4):513–527, dec 1986.
- [18] George Vega Yon. ergm: Exponential Random Graph Models for Small Networks, 2018. R package version 0.1-0.
- [19] George Vega Yon and Paul Marjoram. slurm: A lightweight wrapper for hpc with slurm. The Journal of Open Source Software, 4(39), jul 2019.
- [20] W. N. Venables and B. D. Ripley. Modern Applied Statistics with S. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.
- [21] Stanley Wasserman and Philippa Pattison. Logit models and logistic regressions for social networks: I. An introduction to Markov graphs andp. Psychometrika, 61(3):401–425, 1996.
- [22] Hadley Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.
- [23] Achim Zeileis and Torsten Hothorn. Diagnostic checking in regression relationships. R News, 2(3):7–10, 2002.

A MLE

The estimation process of *ERGMitos* (as pooled models of small networks) is done entirely on R using the `ergmito` R package. While a significant amount of the implementation of the methods described here was done using `Rcpp` [2], a core component of the package is based on `statnet`'s `ergm` R package, and in particular, in the function `ergm.allstats` which does exhaustive enumeration of statistics in a compact way. In general, the estimation process for any list of networks is as follows:

1. Analyze the model to be estimated: Extract the networks from the left-hand-side as specified in the `ergm` package, and calculate the exact statistics using the `ergm.allstats` function.
2. With the full enumeration of statistics, build the joint likelihood function of the model in a compact form (i.e., using the weights instead of the full enumeration of the support of the model). This improves speed when it comes to evaluating the log-likelihood function.
3. Because we are dealing with exact statistics, it is also possible to calculate the exact gradient function. We compute the gradient as follows:

$$\sum_p \nabla l_p(\theta) = s(y_p, X_p)^t - \frac{Q_p^t (W_p^t \circ \exp \{Q_p \theta\})}{\kappa_p} \quad (3)$$

Where $s(y, X)$ is a vector of observed sufficient statistics (usually called target statistics), Q is a matrix of sufficient statistics, in particular, the isomorphic sufficient statistics associated with the model, and W is a vector of frequency weights.

These first three steps carry the most part of the computing time.

4. Finally, the joint log-likelihood is maximized using the BFGS algorithm implemented in the `optim` function in the `stats` package.

The final set of estimates is analyzed separately by another program included in the package. The next section describes the evaluation steps followed by *ERGMito*.

B Evaluation of estimates

After the optimization procedure finalizes, the `ergmito` package performs a series of tests checking the quality of the estimates. In particular, we conduct the following evaluations after every call to the main optimization function:

1. Since the BFGS algorithm, as implemented in the `optim` function from the stats R package, only works with real numbers, we check whether the log-likelihood function increases with changes in the attained value in cases when the theoretical maxima lies in $\pm\infty$. To do this, we increase the magnitude of each estimate by 1.5 and check the value of the log-likelihood function with that change. If an increase in that value is observed, we assume that the maxima for the given parameter equals $\text{sign}(\hat{\theta}_i) \times \infty$.
2. If all parameters turn out to be $\pm\infty$ after this check, the function will send a warning message to the user and the function returns without computing the variance-covariance matrix.
3. If, on the other hand, a fraction of the parameters were switched to $\pm\infty$, the function recalculates the Hessian and the log-likelihood using the value $\text{sign}(\hat{\theta}_i) \times 10^5$. This is done instead of using ∞ , because in most cases using infinite will result in the function being undefined. Again, the function will warn users about this issue.
4. Finally, using the current state of the Hessian matrix, the function will attempt to compute the variance-covariance matrix by inverting the negative of the Hessian. If an error is caught, then the generalized inverse is returned instead [4].

The possible return codes are:

- 00** `optim` converged, no issues reported.
- 01** `optim` converged, but the Hessian is not p.s.d.
- 10** `optim` did not converged, but the estimates look OK.
- 11** `optim` did not converged, and the Hessian is not p.s.d.
- 20** A subset of the parameters estimates was replaced with +/-Inf.
- 21** A subset of the parameters estimates was replaced with +/-Inf, and the Hessian matrix is not p.s.d.
- 30** All parameters went to +/-Inf suggesting that the MLE may not exists.