A semi-parametric statistical test to compare complex networks

Andre Fujita*, Eduardo Silva Lira and Suzana de Siqueira Santos

Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010 - São Paulo - SP, 05508-090 Brazil
*Corresponding author. Email: andrefujita@usp.br

Silvia Yumi Bando
Department of Pediatrics, Faculdade de Medicina, University of São Paulo, Av. Dr. Enéas Carvalho de Aguiar, 647 - Cerqueira César, São Paulo - SP, 05403-000, Brazil

Gabriela Eleuterio Soares
Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, São Paulo, Brazil

AND

Daniel Yasumasa Takahashi
Brain Institute, Federal University of Rio Grande do Norte, Av. Sen. Salgado Filho, 3000 - Lagoa Nova, Natal - RN, 59077-000, Brazil

Edited by: Xiang Li

[Received on 5 February 2019; editorial decision on 8 May 2019; accepted on 16 July 2019]

The modelling of real-world data as complex networks is ubiquitous in several scientific fields, for example, in molecular biology, we study gene regulatory networks and protein–protein interaction (PPI) networks; in neuroscience, we study functional brain networks; and in social science, we analyse social networks. In contrast to theoretical graphs, real-world networks are better modelled as realizations of a random process. Therefore, analyses using methods based on deterministic graphs may be inappropriate. For example, verifying the isomorphism between two graphs is of limited use to decide whether two (or more) real-world networks are generated from the same random process. To overcome this problem, in this article, we introduce a semi-parametric approach similar to the analysis of variance to test the equality of generative models of two or more complex networks. We measure the performance of the proposed statistic using Monte Carlo simulations and illustrate its usefulness by comparing PPI networks of six enteric pathogens.

Keywords: random graph; parameter estimation; model selection; ANOVA; graph spectrum; isomorphism.

1. Introduction

Complex networks are widely used to represent real-world interactions, such as the molecular associations of a specific cell or tissue [1], the functional connectivity among brain regions [2] and the social interac-
tion among individuals of a community [3]. Once we model these associations as complex networks, the common problem is to verify whether networks of different conditions are equal. For example, one may be interested in comparing gene regulatory networks or functional brain networks of people diagnosed with a disease against controls. Alternatively, we could be interested in deciding if the social interaction network of one community is equal to another. Standard computational approaches are based on algorithms to determine isomorphism between graphs, which does not consider the intrinsic randomness present in real-world networks. For example, biological networks may change over time and also among individuals under the same condition. In this case, algorithms that identify isomorphism will falsely discriminate graphs belonging to the same group. Another approach is to analyze the features of the graphs, in particular, centrality measures (e.g., betweenness, eigenvector, closeness, degree) [4, 5]. Usually, the centrality measures are estimated and compared between networks using standard statistical procedures (e.g., Student’s t or Wilcoxon tests). The main drawback with this approach is the fact that networks generated by different models may present the same centrality measure. For example, two Watts–Strogatz (WS) networks generated by different rewiring probabilities present the same average degree centrality because the number of edges does not change. Moreover, graphs generated from the same model can have a vastly different centrality measure (e.g., two scale-free networks). Thus, one solution is to assume that probabilistic processes (complex network models) generate networks and then to test the equality of the models through their parameters.

The first work in this direction was described by Takahashi et al. [6] and then followed by [7–9]. The main limitation of these approaches is that they can only be applied for sets of graphs, that is, it is necessary to observe graph replicates in each population. Nevertheless, this is not the case for the majority of applications, where only one observation (network) per condition is available. For example, we usually observe only one social network replicate, one protein–protein interaction (PPI) network per species, one world wide web and one co-authoring network. Thus, the problem is to test whether the graphs were generated by the same model and parameters, given two or more conditions each one represented by only one network, that is, without replicates. Motivated by this, Ghoshdastidar et al. [10] and Tang et al. [11] proposed statistical tests to compare two networks. The method proposed by Ghoshdastidar et al. [10] decides whether two networks are similar by estimating the average probability of occurrence of triangles for each network. The method proposed by Tang et al. [11] tests the hypothesis that two random dot product graphs have the same underlying distribution by estimating the latent positions used to generate each graph and then comparing these estimations. The main drawbacks of these approaches are the facts that the method proposed by [10] does not provide a p-value (it only outputs whether the networks are ‘equal’ or not), while the method proposed by [11] only works for random dot product graphs. Moreover, none of them can test more than two networks simultaneously. Thus, in this work, we introduce an analysis of variance (ANOVA)-like procedure (for general complex networks) that tests the equality of two or more complex networks without the need for replicates of each sample. Our method is a two-step procedure. We first calculate the spectra of the adjacency matrix and use it to estimate the parameter of the model. Afterward, we use a parametric bootstrap method to build the statistical test. We evaluate its performance by Monte Carlo experiments, benchmark it against the methods proposed by [10, 11] and illustrate its application by comparing six PPI networks of enteric pathogens.

2. Materials and methods

2.1 Test for graphs

A graph (network) \(G = (V, E)\) is composed of a set of \(n\) vertices \(V = \{v_1, v_2, \ldots, v_n\}\) and a set of edges \(E\) that connect pairs of vertices of \(V\). Graphs can be sub-classified to directed (with orientation at the edges) and undirected (without orientation at the edges). Here, we consider only undirected graphs.
Given two graphs \( g_1 \) and \( g_2 \) and the model that generated them (if the models are different the test is not needed), our parametric test for equality of graphs tests whether the parameters used to generate both graphs are equal or not. Formally, let \( \theta_1 \) and \( \theta_2 \) be the parameters used to generate \( g_1 \) and \( g_2 \), respectively, then we test \( H_0 : \theta_1 = \theta_2 \) versus \( H_1 : \theta_1 \neq \theta_2 \).

For the particular case when both \( g_1 \) and \( g_2 \) are generated by an Erdős–Rényi (ER) random graph model (an ER random graph is defined as \( n \) labelled vertices where each pair of vertices \( (v_i, v_j) \) is connected by an edge with probability \( p \)), it is easy to see that the estimator \( \hat{p} = 2 \times \text{number of edges}/(n(n-1)) \) (the number of edges divided by the total number of possible edges) is consistent and can be well approximated by a Gaussian distribution. Therefore, comparing the parameters of two ER random graphs is a direct application of a likelihood ratio test. Following the same logic, if we want to compare two or more ER random graphs, we can design a test similar to the ANOVA. However, for some complex networks models (e.g. WS model), the application of a likelihood ratio test is not any more straightforward because there is no known maximum likelihood (ML) estimator for the parameter of the graph (\( \theta \)) nor an estimator for its variance (\( \sigma^2_{\theta} \)). Thus, for these networks, we propose to estimate the parameter using the semi-parametric method introduced by Takahashi et al. [6] (Section 2.1.1). To estimate its variance, we propose a bootstrap procedure (Section 2.1.2). Finally, we combine both methods and describe an ANOVA-like test for complex networks (Section 2.1.3) that can be applied to compare two or more single realizations of complex networks.

### 2.1.1 Parameter estimation

To estimate the parameter \( \hat{\theta} \) of the complex network model \( \mathcal{M} \) used to generate a network \( g \), we apply the estimation procedure based on the spectral distribution of the complex network [6]. This procedure is described as follows.

Let \( \lambda \) be the set of eigenvalues of the adjacency matrix of the complex network \( g \), \( \delta \) be the Dirac delta function and the brackets ‘\( <> \)’ indicate the expectation with respect to the probability law of the complex network, then the normalized spectral distribution \( \rho_g(\lambda) \) of a network \( g \) is \( \rho_g(\lambda) = \lim_{n \rightarrow \infty} < \frac{1}{n} \sum_{j=1}^{n} \delta(\lambda - \lambda_j/\sqrt{n}) > \). Given a complex network model \( \mathcal{M} \), let \( \theta \) be a real vector containing values for each parameter of \( \mathcal{M} \). If we consider all possible choices for \( \theta \), then the model \( \mathcal{M} \) generates a parametric family of spectral densities \( \rho_\theta \).

Assuming that there is a value \( \theta^* \) that minimizes the Kullback–Leibler (KL) divergence between \( \rho_g \) and \( \rho_\theta \) (\( \text{KL} (\rho_g | \rho_\theta ) = \int_{-\infty}^{\infty} \rho_g(\lambda) \log \frac{\rho_g(\lambda)}{\rho_\theta(\lambda)} d\lambda \)), which is denoted by \( \theta^* \), we have \( \theta^* = \arg\min_{\theta} \text{KL}(\rho_g | \rho_\theta) \) [6, 12].

However, in real applications, the spectral density \( \rho_g \) is unknown. Therefore, we use an estimator \( \hat{\rho}_g \) of \( \rho_g \). To obtain \( \hat{\rho}_g \), we first compute the eigenvalues (\( \lambda \)) of the adjacency matrix of the graph \( g \), and then apply a Gaussian kernel regression with the Nadaraya–Watson estimator [13] for the regularization of the estimator. Finally, we normalize the density to obtain the integral below the curve equal to one. The bandwidth of the kernel can be chosen by the largest eigenvalue minus the smallest eigenvalue divided by the number of bins [14], where the number of bins can be selected by using the Silverman [15] or any other criterion. Therefore, an estimator \( \hat{\theta} \) of \( \theta^* \) is [6, 12]

\[
\hat{\theta} = \arg\min_{\theta} \text{KL} (\hat{\rho}_g | \rho_\theta ).
\] (2.1)

The procedure to estimate \( \hat{\theta} \) is described in Algorithm 2.1.

### 2.1.2 Variance estimation

The asymptotic distribution of \( \theta \) is unknown, thus, it is difficult to analytically estimate its variance (\( \sigma^2_{\theta} \)). Therefore, we use a parametric bootstrap procedure described in Algorithm 2.2.
Algorithm 2.1 Estimation of \( \theta \) via KL divergence

**Input:** network \( g \) and the model \( \mathcal{M} \)

**Output:** the estimated parameter \( \hat{\theta} \)

1. Construct a search grid (an equally spaced set of numbers) with possible values for \( \theta \). For example, for the ER random graph, a grid may assume values of \( \theta = 0, 0.001, 0.002, \ldots, 1 \).
2. For a value of \( \theta \) in the grid, construct \( B \) networks using model \( \mathcal{M} \), and estimate the spectral distribution for each of them (\( \hat{\rho}_\theta^b \), \( b = 1, \ldots, B \)). Then, calculate the average of them (\( \bar{\rho}_\theta = \frac{1}{B} \sum_{b=1}^{B} \hat{\rho}_\theta^b \)). Here, we consider that the analytic spectral distribution of network \( g \) generated by model \( \mathcal{M} \) and parameter \( \theta \) is unknown. Therefore, we estimate it by simulation and consider that for large \( B \), \( \bar{\rho}_\theta \to \rho_\theta \) (\( \hat{\rho}_\theta = \bar{\rho}_\theta \)).
3. Estimate the KL divergence between \( \hat{\rho}_g \) and \( \hat{\rho}_\theta \), that is \( \text{KL}(\hat{\rho}_g | \hat{\rho}_\theta) \).
4. Repeat Steps (2) and (3) for all values of \( \theta \) in the grid.
5. Select the argument \( \theta \) that minimizes the KL divergence between \( \hat{\rho}_g \) and \( \hat{\rho}_\theta \) (see Equation 2.1).

Algorithm 2.2 Estimation of \( \sigma^2_\theta \) via parametric bootstrap

**Input:** model \( \mathcal{M} \) and the estimated parameter \( \hat{\theta} \) of \( g \)

**Output:** the estimated variance of \( \theta \) (\( \hat{\sigma}^2_\theta \))

1. Simulate \( B \) networks (\( g^1, g^2, \ldots, g^B \)) using model \( \mathcal{M} \) and the estimated parameter \( \hat{\theta} \) of \( g \).
2. Estimate the parameters \( \hat{\theta}_1^*, \hat{\theta}_2^*, \ldots, \hat{\theta}_B^* \) of \( g^1, g^2, \ldots, g^B \) using the estimator based on the KL divergence (use Algorithm 2.1).
3. Estimate the bootstrap sample’s mean \( \hat{\theta}^*(.) = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}_b^* \).
4. Estimate the variance of \( \theta \) as \( \hat{\sigma}^2_\theta = \frac{1}{B-1} \sum_{b=1}^{B} (\hat{\theta}_b^* - \hat{\theta}^*(.))^2 \).

2.1.3 ANOVA for graphs

Given the estimators for \( \hat{\theta} \) (Section 2.1.1) and its variance \( \hat{\sigma}^2_\theta \) (Section 2.1.2), we are able to describe an ANOVA-like statistical test to compare two or more graphs.

Let \( g_i, (i = 1, \ldots, m) \) be \( m \) networks with \( n_i \) vertices, and parameters \( \theta_i \), respectively. We would like to test \( H_0: \theta_1 = \theta_2 = \ldots = \theta_m \) versus \( H_1: \text{‘at least one of the parameters (} \theta_i, i = 1, 2, \ldots, m \text{) is different’} \).

Similar to standard ANOVA, the parameters of the networks may be decomposed in: total variation = between networks variation (condition variation—CV) + within variation (residual variation—RV). Thus, the main problem consists of estimating both CV and RV. Algorithm 2.3 shows how to estimate them and also to simultaneously test \( m \) networks in an ANOVA fashion.

2.2 Simulations

We analysed five complex network models, namely ER [16], geometric [17], WS [18], Barabási–Albert (BA) [19] and popularity–similarity (PS) [20]. The construction of networks were performed in \( \mathbb{R} \) using
Algorithm 2.3 ANOVA for networks

**Input**: the complex network model $\mathcal{M}$ and networks $g_1, g_2, \ldots, g_m$

**Output**: the significance of the test ($p$-value)

1. Estimate the parameters $(\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_m)$ of networks $g_1, g_2, \ldots, g_m$ using the KL divergence-based estimator described in Algorithm 2.1.

2. Estimate the variances $\hat{\sigma}^2_{\theta_i}$ for each parameter $\hat{\theta}_i$ ($i = 1, 2, \ldots, m$) using the parametric bootstrap procedure described in Algorithm 2.2 with $B$ bootstrap replicates.

3. Calculate the residual variation as follows: $RV = \sum_{i=1}^{m} (B - 1) \times \hat{\sigma}^2_{\theta_i}$.

4. Let $\bar{\theta} = \frac{1}{m} \sum_{i=1}^{m} \hat{\theta}_i$ be the sample’s mean, then calculate the condition variation as follows: $CV = \sum_{i=1}^{m} (\hat{\theta}_i - \bar{\theta})^2$.

5. The $F$-test rejects $H_0$ at level $\alpha$ if

$$F = \frac{CV/(m - 1)}{RV/(m \times B - m)} > F_{m-1,m \times B-m}(\alpha),$$

where $F_{m-1,m \times B-m}(\alpha)$ is the upper $(100\alpha)$th percentile of the $F$-distribution with $(m - 1)$ and $(m \times B - m)$ degrees of freedom.

The packages igraph and NetHypGeom. Briefly, the construction of these complex networks can be described as follows:

1. **ER random graph**: This graph is defined as $n$ labelled vertices where each pair of vertices is connected by an edge with a given probability $p$.

2. **Geometric random graph**: This graph is constructed by uniformly placing $n$ vertices in $R^d$ space and connecting two vertices by an edge if their distance is smaller than a radius $r$.

3. **WS small-world network**: Let $n$, $n_{ei}$ and $p_w$ be the number of vertices, mean degree and the rewiring probability, respectively. First, connect a ring lattice with $n$ vertices, in which every vertex is connected to its first $n_{ei}$ neighbours. Then, choose a vertex and the edge to a vertex chosen uniformly at random over the ring. Repeat this process by moving clockwise around the ring, considering each vertex in turn until one lap is completed. Next, consider the edges that connect vertices to their second-nearest neighbours. Continue this process outward to more distant neighbours after each lap, until each edge in the original ring has been considered once.

4. **BA random graph**: Start with a small number of ($n_0$) vertices. At each iteration, add a new vertex with $m_1$ ($m_1 \leq n_0$) edges that connect the new vertex to $m_1$ different vertices already present in the graph. To select which vertices the new vertex will connect, assume that the probability that a new vertex will be connected to vertex $v_i$ is proportional to the degree of vertex $v_i$ and the scaling exponent $p_s$ ($(P(v_i) \sim \text{degree}(v_i)^{p_s})$, where degree($v_i$) is the number of adjacent edges of vertex $v_i$ in the current iteration.
(5) PS random graph—start with an empty graph. At each step $t \geq 1$, add a new vertex $v_t$ on coordinate $(r_t, \theta_t)$, where $r_t = \ln t$ and $\theta_t$ is randomly chosen on $[0, 2\pi]$; for each existing vertex $v_s$, $s < t$, update $v_t$ radial coordinate as $r_t(t) = r_s((\gamma - 1) + (\gamma - 2)r_s/(\gamma - 1)$, where $\gamma$ is a parameter on $[2, 3]$ that controls the power-law exponent. Finally, connect $v_t$ to the $m$ closest vertices on the circle according to the hyperbolic distance.

Based on these complex network models, we designed four simulations studies.

2.2.1 Simulation 1 The purpose of this simulation is to evaluate both the control of the rate of false positives and the power of the proposed test when there are three networks (a condition that the ANOVA approach is indeed useful, i.e., when more than two networks are compared). The set of parameters is as follows:

(1) ER model: $p_1 = p_2 = p_3 = 0.5$ (under the null hypothesis) and $p_1 = p_3 = 0.50$ and $p_2 = 0.52$ (under the alternative hypothesis). The number of vertices in each network varied in $n_1 = n_2 = n_3 = 50, 75, 100, 125$.

(2) Geometric model: $r_1 = r_2 = r_3 = 0.3$ (under the null hypothesis) and $r_1 = r_3 = 0.30, r_2 = 0.34$ (under the alternative hypothesis). The number of vertices in each network varied in $n_1 = n_2 = n_3 = 50, 75, 100, 125$.

(3) WS model: $p_1^w = p_2^w = p_3^w = 0.30$ (under the null hypothesis) and $p_1^w = p_3^w = 0.30, p_2^w = 0.35$ (under the alternative hypothesis). The number of vertices in each network varied in $n_1 = n_2 = n_3 = 300, 400, 500, 600$.

(4) BA model: $p_1^w = p_2^w = p_3^w = 1.30$ (under the null hypothesis) and $p_1^w = p_3^w = 1.30, p_2^w = 1.42$ (under the alternative hypothesis). The number of vertices in each network varied in $n_1 = n_2 = n_3 = 700, 800, 900, 1,000$.

(5) PS model: $\gamma_1 = \gamma_2 = \gamma_3 = 2.3$ (under the null hypothesis) and $\gamma_1 = \gamma_3 = 2.30, \gamma_2 = 2.4$ (under the alternative hypothesis). The number of vertices in each network varied in $n_1 = n_2 = n_3 = 700, 800, 900, 1,000$.

2.2.2 Simulation 2 The purpose of this simulation is to evaluate both the control of the rate of false positives and the power of the proposed test on the comparison of networks of different sizes, that is, different number of vertices. The set-up of this experiment is similar to simulation 1, but one of the networks present a different size (different number of vertices):

(1) ER model: $p_1 = p_2 = p_3 = 0.5$ (under the null hypothesis) and $p_1 = p_3 = 0.50$ and $p_2 = 0.52$ (under the alternative hypothesis). The number of vertices are $n_1 = n_3 = 125$ and $n_2$ varying in $n_2 = 50, 75, 100, 125$.

(2) Geometric model: $r_1 = r_2 = r_3 = 0.3$ (under the null hypothesis) and $r_1 = r_3 = 0.30, r_2 = 0.34$ (under the alternative hypothesis). The number of vertices is $n_1 = n_3 = 125$ and $n_2$ varied in $n_2 = 50, 75, 100, 125$.

(3) WS model: $p_1^w = p_2^w = p_3^w = 0.30$ (under the null hypothesis) and $p_1^w = p_3^w = 0.30, p_2^w = 0.35$ (under the alternative hypothesis). The number of vertices is $n_1 = n_3 = 600$ and $n_2$ varied in $n_2 = 300, 400, 500, 600$. 
A STATISTICAL TEST FOR COMPLEX NETWORKS

(4) BA model: $p^*_1 = p^*_2 = p^*_3 = 1.30$ (under the null hypothesis) and $p^*_1 = p^*_3 = 1.30, p^*_2 = 1.42$ (under the alternative hypothesis). The number of vertices is $n_1 = n_3 = 1000$ and $n_2$ varied in $n_2 = 700, 800, 900, 1000$.

(5) PS model: $\gamma_1 = \gamma_2 = \gamma_3 = 2.3$ (under the null hypothesis) and $\gamma_1 = \gamma_3 = 2.3, \gamma_2 = 2.4$ (under the alternative hypothesis). The number of vertices is $n_1 = n_3 = 1000$ and $n_2$ varied in $n_2 = 700, 800, 900, 1000$.

2.2.3 Simulation 3 The purpose of this simulation is to evaluate both the control of the rate of false positives and the power of the proposed test when the true random graph model is unknown. Two BA random graphs are generated with the following set of parameters: $p^*_1 = p^*_2 = 1.0$ (under the null hypothesis) and $p^*_1 = 1.0$ and $p^*_2 = 1.3$ (under the alternative hypothesis). The number of vertices in each network varied in $n_1 = n_2 = 50, 100, 200$. Since, we assume that the true model is unknown, it is necessary to infer it. To this end, we use the model selection approach described in Takahashi et al. (2012). For the case the model selection approach selects two different models for the two graphs, we set the $p$-value equals to zero and do not carry out the proposed test.

2.2.4 Simulation 4 The purpose of this simulation is to benchmark the developed method against the approaches proposed by Tang et al. [11] and Ghoshdsticar et al. [10]. Since the methods proposed by Tang et al. [11] and Ghoshdsticar et al. [10] can only compare two networks of equal sizes, we carried out simulation 1 by considering only graphs 1 and 2.

For each simulation study (Sections 2.2.1–2.2.4) and each network’s size $n$ (number of vertices), we repeated the experiments 500 times. Then, to evaluate the performance of the proposed test under different conditions and network’s sizes, we constructed receiver operating characteristic (ROC) curves. For simulation 3, since the method proposed by Ghoshdsticar et al. [10] does not provide a $p$-value (its output only indicates whether the graphs are equal or different), we set the $p$-value threshold for the methods proposed by [11] and ours to 0.05 and counted the number of times each method identified different networks. A ROC curve is a bi-dimensional plot with one minus the specificity at the $x$-axis and the sensitivity at the $y$-axis. In our case, the $p$-value threshold used in the test is at the $x$-axis and the proportion of rejected null hypotheses at the $y$-axis. Under the null hypothesis, we expect that the proportion of falsely rejected null hypotheses is the same as set by the $p$-value threshold. In other words, we expect a ROC curve lying at the diagonal. Under the alternative hypothesis, we expect that the curve be above the diagonal, that is, the test rejects $H_0$ more than at random. The farther (above) the ROC curve is from the diagonal, the greater is the power of the test.

2.3 Real data application

We selected six representative enteric pathogens, namely, (i) Campylobacter jejuni NCTC11168, (ii) Escherichia coli O157:H7, (iii) Listeria monocytogenes EGDe, (iv) Salmonella enterica LT2, (v) Shigella flexneri 2a str.301 and (vi) Yersinia enterocolitica 8081. These species belong to Enterobacteriaceae and the strains selected here are frequently associated with gastroenteritis. Their respective PPI networks were downloaded from string-db.org. Vertices represent the proteins, and edges represent the interactions among them. We considered only interactions with experimental evidences, that is, score $> 0$ (we discarded predicted or putative interactions). Descriptive details of the networks are summarized in Table 1.
Table 1: Summary of the number of vertices (proteins) and edges (interactions) for each species

<table>
<thead>
<tr>
<th>Species (strain)</th>
<th>Number of vertices (n)</th>
<th>Number of edges (interactions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C. jejuni (NCTC11168)</td>
<td>1398</td>
<td>16 135</td>
</tr>
<tr>
<td>E. coli (O157:H7)</td>
<td>3295</td>
<td>50 838</td>
</tr>
<tr>
<td>L. monocytogenes (EGDe)</td>
<td>1760</td>
<td>27 180</td>
</tr>
<tr>
<td>S. enterica (LT2)</td>
<td>2908</td>
<td>42 035</td>
</tr>
<tr>
<td>S. flexneri (2a str.301)</td>
<td>2980</td>
<td>38 597</td>
</tr>
<tr>
<td>Y. enterocolitica (8081)</td>
<td>2592</td>
<td>34 759</td>
</tr>
</tbody>
</table>

3. Results and discussion

Since our approach is based on the $F$-test, we checked whether the probability distribution of the estimator based on the KL divergence can be approximated by a normal distribution. The derivation of the asymptotic distributions for general random graphs are not trivial and no general results are described in the literature. Thus, we analysed it computationally. By Monte Carlo simulations, we verified that the KL divergence-based estimator can be indeed approximated by a normal distribution (for further details, refer to the Appendix).

Then, to verify if the proposed method is able to control the type I error and also correctly identify graphs generated by different parameters, we performed the simulations described in Section 2.2 and constructed the respective ROC curves.

Results for the simulation described in Section 2.2.1 are shown in Fig. 1. Notice that, under the null hypothesis (when networks $g_1$, $g_2$ and $g_3$ are generated with the same parameters), the proposed test effectively controls the type I error. In practice, although the KL divergence-based estimator may not follow perfectly a normal distribution, its approximation is sufficient to our test effectively controls the type I error. By analysing the results under the alternative hypothesis (when networks $g_1$, $g_2$ and $g_3$ are generated by different parameters), we observe that the larger is the graph, the greater is the power of the test.

During the simulations, we noticed that for our proposed test be able to effectively control the type I error, WS, BA and PS networks should be relatively large ($n > 300$ for WS and $n > 700$ for BA and PS). This is due to the precision of the parameter estimate. This is consistent with [12] in which the authors show that the convergence rate of the parameter estimate for WS and BA is slower than for the ER and geometric random graph models.

For the special case when the networks are generated by an ER model, there is a ML estimator for $\theta$ that is consistent and efficient (see Section 2.1). Thus, we compared the performance of the proposed test using the KL divergence-based estimator (Section 2.1.1) versus the ML estimator for $m = 2$ (two graphs) and $m = 3$ (three graphs) ER graphs. Since the area under the ROC curve (AUC) represents the power of the test, we estimated the AUCs and respective 95% confidence intervals for each ROC curve (Fig. 2). By analysing Fig. 2, we note that the powers of the tests are equivalent for both estimators (ML and KL), independent of the graph’s size ($n$).

To evaluate the performance of the test on networks of different sizes (number of vertices), we performed the experiment described in Section 2.2.2 and show the obtained ROC curves in Fig. 3. By analysing Fig. 3, we note that, under the null hypothesis, our test, in fact, controls the type I error even when the networks’ sizes are different. We also observe that the more similar the networks’ sizes are, the
greater is the power of the test, as expected. This experiment demonstrates that, in fact, the proposed test controls the rate of false positives and also is able to identify whether the networks were generated by different parameters even when the sizes are different.

To evaluate the performance of the test when the true network model is unknown, we carried out the experiment described in Section 2.2.3 and show the obtained ROC curves in Fig. 4. By analysing Fig. 4, we note that, under the null hypothesis, for small graphs \( n = 50 \), the control of the type I error is not effective. It can be explained by the fact that in approximately 12% of the cases, the model selection
approach selected the ER model instead of the true BA model. For larger graphs, that is, $n = 100$ and 200, the number of times that the model selection approach wrongly selected the ER model decreases to 1 and 0, respectively. Consequently, we see better control of the type I error.

Then, we benchmarked our proposed test against two methods: the methods proposed by Tang et al. [11] and Ghoshdastidar et al. [10] (simulation described in Section 2.2.4). Results described in Fig. 5 show that for the ER model, all methods indeed control the type I error (since the $p$-value threshold was set to 5%, the expected proportion of rejected null hypotheses is 5%). However, for the geometric, WS, BA and PS, only our proposed method effectively controlled the rate of false positives. Notice that the method proposed by Tang et al. rejected much more than the expected, while the method proposed by Ghoshdastidar et al. rejected proportionally to the network’s size for the geometric model, and did not reject anything for the WS, BA and PS random graph models. These behaviours explain the high power of the method proposed by Tang et al.—but only because it rejects most tests—and the lack of power for the method proposed by Ghoshdastidar et al. under the alternative hypothesis ($H_1$). It is important to mention that the lack of control of the type I error for the methods proposed by Tang et al. and Ghoshdastidar et al. are as expected because they were not designed for these random graph models. However, even for the ER model, which their methods work correctly, our proposed method presented greater statistical power.

Finally, we applied the proposed test to six PPI networks described in Section 2.3. As largely described in the literature, we assumed the BA model [1] and tested its scale exponent (parameter power of the R function barabasi.game). Our test suggests that there is statistical evidence to reject the null hypothesis, that is, that there is at least one PPI network that is generated with a different parameter ($p < 0.001$). Then, to identify the PPI networks that are different, we tested them in a pairwise manner. The corrected $p$-values for false discovery rate (FDR) [21] of the pairwise tests are described in Table 2.
By analysing Table 2, there is no statistical evidences to discriminate the networks’ structures of *E. coli* and *S. flexneri* \((p = 0.924)\). In fact, these two species are known to share very similar biochemical and phenotypical characteristics. Moreover, they have been considered genetically close [22]. The analysis between *C. jejuni* and *S. enterica* also did not show statistical evidence \((p = 0.791)\) to state that they are different. It is interesting to note that poultry is a common reservoir of these two pathogens [23].
Fig. 4. Application of the ANOVA-like test when the network model is unknown (simulation described in Section 2.2.3). The $x$-axis represents the $p$-value threshold, and the $y$-axis represents the proportion of rejected null hypothesis in 500 repetitions. The different types of lines (solid and dashed) represent the ROC curves obtained for different sizes of networks. The $H_0$ represents the ROC curves when the two networks are generated with the same parameters ($p_1 = p_2 = 1.0$). The $H_1$ represents the ROC curves when two networks are generated with different parameters ($p_1 = 1.0$ and $p_2 = 1.3$). Since, we assume that the network models are unknown, we inferred it by using a model selection approach. The lack of control of the type I error under $H_0$ for small graphs ($n = 50$) is due to the model selection approach that, in approximately 12% of the times, it wrongly selected the ER model instead of the correct BA model.

Conversely, the PPI network of *Y. enterocolitica* seems to be statistically distinct of others PPI networks ($p < 0.001$), namely, *C. jejuni*, *L. monocytogenes* and *S. enterica*. It is curious because during pathogen coinfection, *Y. enterocolitica* inhibits *S. enterica* and *L. monocytogenes* infection into host-cells through diminishing phagocytosis induction to bacterial uptake and guarantee bacterial survival [24].

3.1 Final considerations

For some specific complex network models, such as the k-regular model [25], the proposed test is unnecessary. A k-regular network [25] is a network where every vertex has the same degree $k$ (number of adjacent vertices). Consequently, the variance of the degree is zero, and the test would consist of determining the degree of one vertex of each network and directly verifying the equality of them.

Here, we assumed that the model that generated the networks is known. However, in real applications, the network model is rarely known. In this case, we propose to first identify the model that best fits the data by using the model selection approach described in [6]. If the selected models are equal, then our proposed method can be used to test the parameters. Otherwise, the networks are considered different.

One limitation of the proposed approach is the fact that this idea can only be applied for undirected complex network models. Notice that our approach requires a parameter estimation procedure, which currently can be applied only to undirected networks. Some preliminary results suggest that the same parameter estimator can be applied to directed graphs whether we consider the imaginary number as a second dimension. However, further studies are still required.

For network models with several parameters, a multivariate ANOVA for networks may be more appropriate and useful. Based on the ideas described here, we believe that a generalization of the proposed test for a multivariate version would be straightforward.

Here, we showed that our proposal has greater statistical power than the methods proposed by Ghoshdastidar *et al.* [10] and Tang *et al.* [11] for the specific five complex network models (ER, geometric, WS,
Fig. 5. Comparison of the methods. Since the methods proposed by Tang et al. [11] and Ghoshdastidar et al. [10] are only able to compare two networks with equal sizes, we carried out the set-up of simulation 1 (see Section 2.2.1) by considering only two networks, namely networks 1 and 2. To discriminate whether there is or not differences between the networks, we set the $p$-value threshold for the methods proposed by Tang et al. [11] and ours to 0.05. The method proposed by Ghoshdastidar et al. does not provide a $p$-value; it just outputs whether the graphs are different or not. The bars represent the proportion of times the methods rejected the null hypotheses. The error bars represent the 95% confidence intervals. For the ER model, all the evaluated methods were able to effectively control the rate of false positives and also discriminate the networks when they were in fact generated by different parameters. The methods proposed by Tang et al. [11] and Ghoshdastidar et al. [10] do not control the rate of false positives under the null hypothesis ($H_0$) for geometric, WS, BA and PS models. Notice that it is expected a proportion of rejected null hypotheses of 5%, but the method proposed by Tang rejected much more than the expected, while the method proposed by Ghoshdastidar et al. rejected proportionally to the network’s size for the geometric model, and did not reject anything for the WS, BA and PS models. These behaviours explain the pseudo high power of the method proposed by Tang et al. and the lack of power for the method proposed by Ghoshdastidar et al. under the alternative hypothesis ($H_1$).
Table 2  p-Values obtained from the application of the proposed test on six PPI networks in a pairwise manner. p-Values are corrected for multiple tests by the FDR approach [21]. Significant p-values at a threshold of 5% are in bold.

<table>
<thead>
<tr>
<th></th>
<th>E. coli</th>
<th>L. monocytogenes</th>
<th>S. enterica</th>
<th>S. flexneri</th>
<th>Y. enterocolitica</th>
</tr>
</thead>
<tbody>
<tr>
<td>C. jejuni</td>
<td>0.141</td>
<td>0.286</td>
<td>0.791</td>
<td>0.162</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>E. coli</td>
<td>0.319</td>
<td>0.174</td>
<td>0.924</td>
<td>0.064</td>
<td></td>
</tr>
<tr>
<td>L. monocytogenes</td>
<td>0.372</td>
<td>0.372</td>
<td></td>
<td></td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>S. enterica</td>
<td></td>
<td>0.204</td>
<td></td>
<td></td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>S. flexneri</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.064</td>
</tr>
</tbody>
</table>

BS and PS) evaluated here. However, if the complex network model presents the number of parameters of the order of $n^2$ (e.g. random dot product graphs), our method is not able to estimate adequately all the parameters, and consequently, its performance would be impaired.

The proposed ANOVA test for networks is implemented in R, package statGraph—function sp.anogva and can be downloaded from the R website (https://www.r-project.org).

4. Conclusion

We presented a semi-parametric ANOVA-like method to test whether two or more networks are generated by the same set of parameters. Our test only requires the observation of a single realization from each network. We showed that the method is simple and robust for the main types of random graphs considered in the literature. The proposed test is model-based, therefore it requires that the random graph model is known or can be reliably inferred from the data. When the random graph follows a distribution that is not well modelled by known random graphs, our procedure cannot guarantee to successfully distinguish the networks. In this case, a fully non-parametric method might be required. Our work is based on extensive simulations but still lacks theoretical guarantees in general. Fast progress on the mathematical theory of random graphs has been made in recent years, especially for graphs with strongly correlated edges. Hence, we expect that some of the theoretical problems can be adequately solved in near future.

Funding

São Paulo Research Foundation (2015/21162-4 to S.S.S., 2015/01587-0, 2018/17996-5 and 2018/21934-5 to A.F.); CNPq (304876/2016-0 to A.F.); CAPES (Finance Code 001 to E.S.L., G.E.S. and S.S.S.); Alexander von Humboldt Foundation to A.F.; Newton Fund/The Academy of Medical Sciences to A.F.; European Research Council to A.F.; and Pew Latin American Fellowship to D.Y.T.

References

Appendix

Our proposed method is based on the F-test, which assumes that the used parameter estimator follows a normal distribution. Thus, we designed a simulation study to verify whether the estimated parameters by the KL divergence-based estimator can be approximated by a normal distribution. The set-up is as follows:

1. ER model: $p = 0.5$ and $n = 125$.
2. Geometric model: $r = 0.3$ and $n = 125$. 
(3) WS model: \( p = 0.30 \) and \( n = 600 \).

(4) BA model: \( p = 1.5 \) and \( n = 5000 \).

(5) PS model: \( p = 2.5 \) and \( n = 5000 \).

Each network was simulated 500 times and their parameters estimated by using the KL-based estimator.

Figure A.1 illustrates the probability distributions of the estimated parameters. Notice that all of them can be approximated by a normal distribution and the estimated parameter converges to the true parameter.
Fig. A.1. Probability distribution of the estimated parameters by the KL divergence-based estimator. The x- and y-axes represent the estimated parameter and its frequency, respectively. The probability distributions were constructed based on 500 repetitions. The vertical red dashed line represents the true parameter value. Notice that for the evaluated complex network models, the KL-based estimator converges to the correct value.