Victor Chavauty Villela

Statistical Methods for Directed Graphs Based on the Graph Spectrum

Versão Corrigida

Dissertação de mestrado apresentada ao Programa Interunidades de Pós-Graduação em Bioinformática da Universidade de São Paulo para obtenção do título de Mestre em Ciências.

Área de Concentração: Bioinformática
Orientador: Prof. Dr. André Fujita

São Paulo
2024

Ficha Catalográfica elaborada eletronicamente pelo autor, utilizando o programa desenvolvido pela Seção Técnica de Informática do ICMC/USP e adaptado para a Divisão de Biblioteca e Documentação do Conjunto das Químicas da USP

```
        Chavauty, Victor
C512s      Statistical Methods for Directed Graphs Based on
        the Graph Spectrum / Victor Chavauty. - São Paulo,
        2024.
           37 p.


        Dissertação (mestrado) - Programa Interunidades de
        Pós-Graduação em Bioinformática - USP, Instituto de
        Matemática e Estatística.
           Orientador: Fujita, André


           1. Estatística de grafos. I. T. II. Fujita,
        André, orientador.
```

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

Victor Chavauty Villela. **Métodos estatísticos baseados no espectro para grafos dirigidos**. Dissertação (Mestrado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2024.

Grafos são frequentemente utilizados para modelar mecanismos complexos em diversas áreas de estudo, como redes sociais (social networks), conectividade entre regiões do cérebro, ou interações proteína-proteína. No entanto, devido a complexidade de sua estrutura, métodos estatísticos padrão não são suficientes para encontrar correlações entre populações de grafos. Em um trabalho recente por Takahashi et al. (2012) foi sugerido que o espectro do grafo é uma boa caracterização de sua estrutura, e diversos métodos estatísticos foram construídos baseado nessa ideia. Entretanto, esses métodos dependem dos autovalores do grafo terem valor real, o que não é satisfeito quando grafos são dirigidos. Neste trabalho estendemos estes resultados para grafo dirigidos utilizando a distribuição de autovalores complexa como base. Assim, desenvolvemos métodos de estimação de parâmetros para modelos de grafos aleatórios, uma seleção de modelos, um teste estatístico para comparar duas ou mais populações de grafos, um teste de associação entre grafos e variáveis de interesse, e um algoritmo de agrupamento.

Palavras-chave: Correlação de redes. Estatística de grafos. ECoG.

Abstract

Victor Chavauty Villela. **Statistical Methods for Directed Graphs Based on the Graph Spectrum**. Thesis (Master's). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2024.

Graphs are often used to model diverse, complex phenomena in many fields, such as social networks, brain region connectivity, or protein-protein interaction. However, due to the complexity of their structure, standard statistical methods are insufficient in searching for a correlation between populations of graphs. In a recent paper by Takahashi et al. (2012), they suggested that the graph spectrum is a good fingerprint of the graph's structure, and they developed several statistical methods based on this feature. These methods, however, rely on the distribution of the eigenvalues of the graph being real-valued, which is false when graphs are directed or weighted. In this thesis, we extend their results to directed graphs by working with the distribution of complex eigenvalues instead. Then, we constructed a parameter estimator, a model selection approach, a statistical test to compare two populations of graphs, a test for association between a graph and variables of interest, and a clustering algorithm.

Keywords: Network Correlation. Graph Statistics. ECoG.

Lista de Figuras

Lista de Tabelas

Sumário

Apêndices

Anexos

Capítulo 1

Introduction

Graphs are often used to model the connection between things. Examples in the literature include brain connectivity (Bullmore e Sporns, 2009), social interactions (Scott, 2012), molecular interactions (Barabasi e Oltvai, 2004), or even the inter-regulation of genes (Alon, 2006) (Andre Fujita *et al.*, 2020).

Suppose, for example, we want to establish how a treatment affects different patients with hormonal issues. We have three groups of patients, and each group receives a different medication. By looking at the predicted gene network linked to how cells work, can we tell which treatment was given?

Traditional computational methods to find similarities between graphs or sub-graphs do not work when graphs present intrinsic randomness. Unfortunately, randomness is intrinsic to biological data, which makes it impervious to traditional computational methods (Siqueira Santos *et al.*, 2014; Andre Fujita *et al.*, 2020). Instead, more common approaches include using graph characteristics like the number of nodes, edges, or centrality measures. For example, one may apply traditional statistical methods to a centrality measure extracted from the graphs (Siqueira Santos *et al.*, 2014).

Although centrality measures are helpful, they might not show the vast differences between graphs. For example, take two graphs made using the Watts-Strogatz model, each with a different rewiring probability. Although they were created with a different rewiring parameter, they present the same average degree centrality measure because they have the same number of connections (Andre Fujita *et al.*, 2020).

In 2012, Takahashi et al. (Takahashi *et al.*, 2012) proposed that the graph spectrum is a good feature for describing the graph structure. They used the Kullback-Leibler and Jensen-Shannon divergences between spectral distributions to measure the distance between graphs. Using this concept, they constructed tools for

- a statistical test to compare two sets of graphs;
- a parameter estimator for random graph models; and
- a model selection approach.

Recently, these ideas have been used to create a concept of correlation (Daniel Yasumasa Takahashi *et al.*, 2017) / causality (Ribeiro *et al.*, 2021) between graphs and spectrum-based clustering algorithms for complex networks (Ramos *et al.*, 2023).

Unfortunately, those methods are limited to undirected graphs with real eigenvalues. Instead, I have extended them to directed graphs by examining the distribution of complex eigenvalues.

In summary, I have implemented methods for:

- Estimating parameters for random graph models;
- Selecting the model that best fits the observed graph;
- Comparing between two or more populations of graphs (Chavauty *et al.*, s.d.);
- Testing the correlation between graphs and other factors;
- Grouping graphs using k-medoids clustering.

To showcase a demonstration of graph spectrum, I have also applied one of these methods to a dataset consisting of ECoG data of a monkey under different stages of anesthesia (Chavauty *et al.*, s.d.).

Capítulo 2

Materials

2.1 Graphs

A graph G consists of a pair (N, E), where N is a set of nodes, and E is a set of edges connecting a pair of nodes of G.

We will refer to this graph as weighted if every edge between two nodes i and j of a given graph is associated with a complex value $e_{i,j} \in \mathbb{C}$. In contrast, in non-weighted graphs, an edge between two nodes i and j will assume 1 if i and j are connected or 0 otherwise.

If for every pair of nodes i and j, the edges $e_{i,j}$ and $e_{j,i}$ (connecting i to j and j to i) are equal, then we will call that graph undirected. Otherwise, it is directed.

A graph G's adjacency matrix is defined as $\mathbf{A}_G = (e_{i,j})_{i,j=1,\ldots,n}$, where $e_{i,j}$ is the value associated with the edge connecting node i and node j.

We define the spectrum of a graph G as the set of eigenvalues of its adjacency matrix $\mathbf{A}_G$. If G is directed, its adjacency matrix is non-symmetrical. Therefore, its eigenvalues are complex-valued. If G is undirected, its adjacency matrix is symmetrical, and its eigenvalues are real-valued.

2.2 Spectral distribution

We define a random graph g as a family of graphs whose members are generated by a probability law. For example, the Erdös-Rényi random graph is generated by creating n nodes and connecting two nodes with a uniform probability p.

The complex Dirac delta is defined as the measure $\delta_{\mathbb{C}}$ satisfying for every compactly supported continuous function f:

$$\int_{\mathbb{C}} f(x)\delta_{\mathbb{C}}\{dx\} = f(0).$$

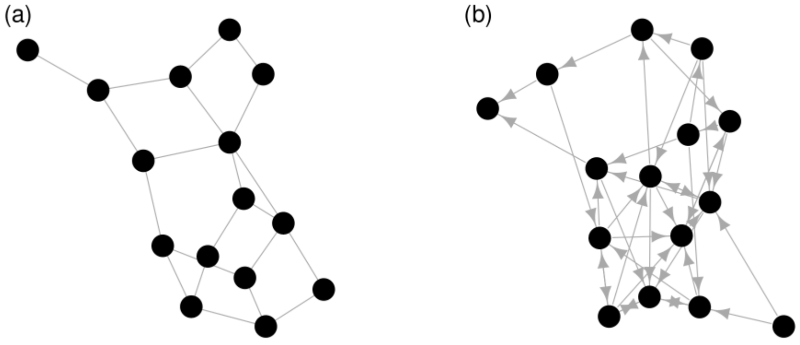


Figura 2.1: *(a) An undirected graph on the left (b) A directed graph, where arrows indicate directionality.*

An equivalent construction of the Dirac delta is the product of the 1-dimensional Dirac delta in two variables, representing the real and the imaginary variables:

$$\delta_{\mathbb{C}}(a + bi) = \delta(a)\delta(b).$$

Suppose g is a directed random graph generated by some probability law. The complex eigenvalues of its adjacency matrix Δ form random vectors. Let brackets $\langle\rangle$ indicate expectations concerning the probability law. In this scenario, we define the spectral distribution of the random graph g as

$$\rho_g(\lambda) = \lim_{n\to\infty} \langle \frac{1}{n} \sum_{j=1}^{n} \delta_{\mathbb{C}}(\lambda - \frac{\lambda_j}{\sqrt{n}}) \rangle.$$

The distribution of ρ_g can be used as a fingerprint of the random graph g (Chavauty *et al.*, s.d.).

2.2.1 Calculating the graph spectrum

The spectral density ρ_g is generally not known. This motivates us to construct an estimator $\hat{\rho}_g$.

To construct this estimator, we can follow a similar procedure for the undirected case (Andre Fujita *et al.*, 2020).

First, we compute the eigenvalues $\lambda_1, \dots, \lambda_n$ of the graph's adjacency matrix and apply a multivariate kernel regression (Tran Duong, 2007). Dividing the resulting 2-dimensional surface by the volume under the curve ensures that the result is a probability distribution.

In Figure 2.2, we can see the difference between the spectral distribution of directed and undirected graphs. By analyzing directionality, we also increase the dimensionality of the space in which the eigenvalues reside, which impacts algorithm performance.

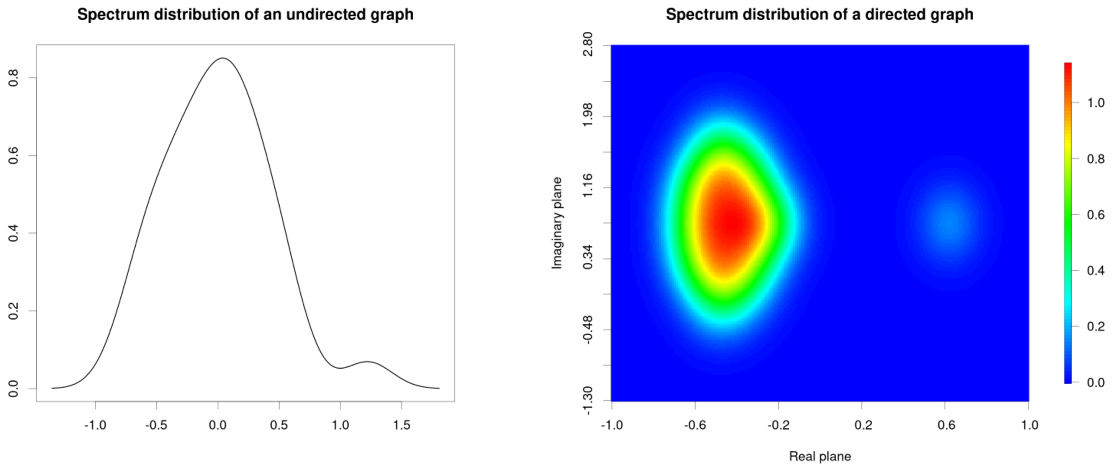


Figura 2.2: *Eigenvalues of undirected graphs are distributed on the real line, whereas eigenvalues of directed graphs are distributed on the complex plane.*

2.3 Entropy and Statistical Distance

2.3.1 Entropy

Let G be a graph and ρ_G be its spectrum. Following the usual convention that $0log(0) = 0$, we can define the entropy of the graph as

$$H(G) = H(\rho_G) = -\int_{\mathbb{C}} \rho_G(\lambda) log \rho_G(\lambda) d\lambda$$

This is also known as differential entropy (Cover e Thomas, 2006). The spectral entropy can be seen as a measure of a form of uncertainty that is associated with the random graph. For example, Takahashi et. al. (Takahashi *et al.*, 2012) showed that the maximum entropy for the Erdos-Renyi graph is achieved when edges are connected with probability 0.5.

2.3.2 Statistical Distance

The spectrum distribution is the distribution of complex eigenvalues of a graph model. This spectrum distribution can be used as a fingerprint of the model so that by comparing the spectrum of two different random graph models, we can establish a certain distance between them (Takahashi *et al.*, 2012, Van Mieghem, 2011). Similarly, we can compare the spectrum of a graph to the spectrum distribution of a random graph model and obtain a measure of how far apart the graph is from being generated from that specific model. We are going to rely on distance functions that have been used literally.

An example of those is the Kullback-Leibler (KL) (MacKay, 2003) divergence, which compares between statistical distributions. For two probability densities, p, and q, the Kullback-Leibler divergence is defined as

$$KL(p,q) = \int_{\mathbb{C}} p(x) \log \left(\frac{p(x)}{q(x)} \right) dx$$

It is a powerful statistical pseudo-distance measuring how a probability distribution differs from a second distribution. It is not strictly a distance in the context of metric spaces.

Alternatively, we can use the classic L_p distance defined as

$$L_p(p,q) = \int_{\mathbb{C}} \|p(x) - q(x)\|_p dx$$

2.4 Random graph models

It is often impossible to establish how a graph was formed when dealing with biological data. Besides, it is difficult to establish whether two graphs are similar simply by analyzing their structures. Thus, one idea is to imagine these graphs resulting from a probabilistic model with a set of parameters. There are several undirected random graph models. Each random graph model has its advantages and disadvantages when modeling real-life phenomena. In particular, we highlight three random graph models that have been used in academia: Erdős–Rényi (ER) (Erdős e Rényi, 1959), Watts–Strogatz (WS) (Watts e Strogatz, 1998) and the Barabási–Albert (PA) (Barabási e Albert, 1999).

Directed models

Unfortunately, models for directed graphs are not as prevalent as the ones for undirected graphs. We have developed a general extension of any directed random graph model. This extension allows us to generate various directed random graphs on which we plan to perform simulations. The description of the method is as follows:

Given a random model r with a parameter p, we extend this model as follows. Let p_1 and p_2 be two parameters for model M. Then

1. Generate a graph G_1 with parameter p_1 and construct its adjacency matrix.
2. Generate a graph G_1 with parameter p_2 and construct its adjacency matrix.
3. Generate a matrix $\mathbf{A}$ whose upper triangular is the same as of G_1 and whose lower triangular is the same as of G_2.
4. Generate a graph G with adjacency matrix $\mathbf{A}$.

The parameters p_1 and p_2 control the network's inner and external connections, respectively, represented on the upper and lower triangles of the graph's adjacency matrix.

In the scenario in which $p_1 = p_2$, the resulting graph is still directed due to the random element of the graph generation process.

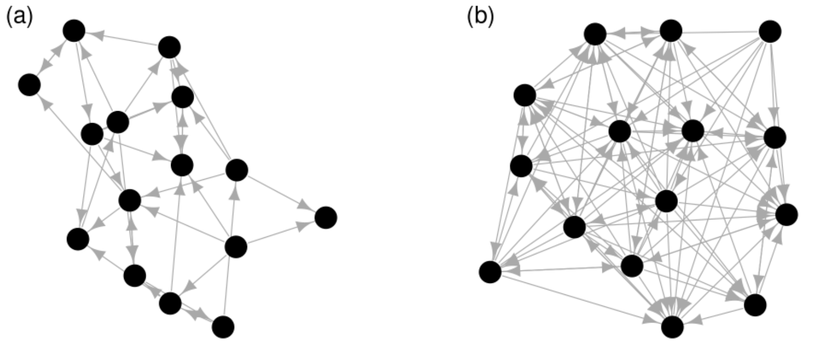


Figura 2.3: *(a) Directed Erdős-Rényi graph generated using parameters* $(0.2, 0.2)$ *(b) Directed Erdős-Rényi graph generated using parameters* $(0.2, 0.8)$.

Capítulo 3

Methods

Graphs face several distinct problems. Using the graph spectrum as a fingerprint of a graph allows us to solve several of these problems. The following is a list of the methods that have been implemented.

3.1 Parameter Estimation

A wide range of random graph models can be used to model natural phenomena. Each model accepts a set of parameters as input and outputs a graph. Different random graph models are suitable for modeling different biological or physical behaviors.

However, suppose we have a graph G, which we are certain comes from a specific random model M. How can we establish which set of parameters $p = (p_1, \dots, p_k)$ was used to generate G?

Let p_0 be a set of valid parameters for the random graph model M. Then, there exists a set of graphs, M_{p_0}, corresponding to the graphs that can be generated by M under that specific parameter. We can estimate the graph spectrum of this set. Let ρ_{p_0} be this spectrum. Let D be a notion of distance (such as L_1, L_2, or KL). Then, suppose ρ_G is the graph spectrum of the original graph G. In that case, we can calculate the distance from this graph to the graph model under the parameter ρ_{p_0} by $D(\rho_G, \rho_{p_0})$

Under this notation, we say that the ideal parameter p is the one that minimizes the distance $D(\rho_G, \rho_p)$.

However, we still need to estimate the spectral distribution of the random graph model under a parameter p. We construct this estimator by averaging the spectral distribution of N samples of the random model M and parameter p

$$\hat{\rho}_p(\lambda) = \frac{1}{N} \sum_{i-1}^{N} \rho_{G_i}(\lambda)$$

We then estimate the parameter as

$$\hat{p} = \arg\min_{p} D(\rho_G, \hat{\rho_p}) \tag{1}$$

Calculating this estimate is simply a matter of minimizing the function $D(\rho_G, \hat{\rho_p})$.

Grid Search: The grid search consists of dividing the search space in a grid. The search then iterates through every point in the grid, keeping track of the point that minimizes the function. With a sufficiently small grid spacing, this method effectively finds the correct parameter but with a high computational cost. This cost is exponential based on the number of parameters of the random graph model. Therefore, it becomes unfeasible if we attempt to estimate several parameters simultaneously.

Ternary Search: The ternary search is a divide-and-conquer algorithm that efficiently locates an unimodal function's maximum or minimum point. It recursively narrows the search interval into three segments and compares function values at two points within those segments. It is much more efficient than the grid search but may not work for all models. If the distance function is not unimodal, then the search will fail. To minimize this risk, our simulations show that working with L_p distances is better than KL.

Figure 3.1 shows the difference between grid search and ternary search.

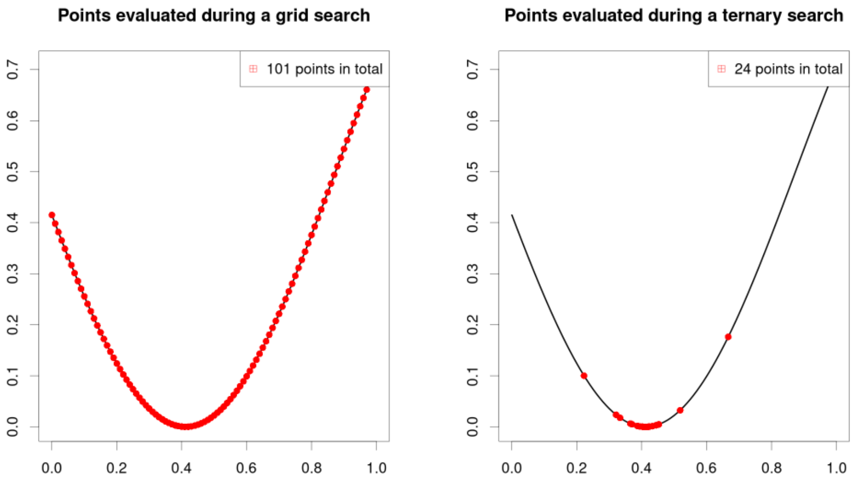


Figura 3.1: *Difference between a grid search and a ternary search: The black function illustrates a common distance function we aim to minimize. The red dots represent points requiring evaluation for the method to estimate the minimum. The grid search evenly divides the function into several chunks, necessitating a total of 101 points for evaluation. In contrast, the ternary search only demands the evaluation of 24 points. In parameter estimation, evaluating distance functions involves constructing multiple graphs from a given model, estimating their spectral distribution, and using it as input in a distance function. Therefore, it is important to minimize this evaluation whenever possible.*

3.2 Model Selection

Given a graph G, several random graph models $M_1, \dots, M_N$ can be ranked by their Kullback-Leibler divergence. Models with low KL divergence are better for generating G (Takahashi *et al.*, 2012).

Let $M_1, \dots, M_N$ be N different random graph models with, and let $\hat{p}_1, \dots, \hat{p}_N$ be their respective estimators given by equation (1). Recall that each parameter $\hat{p}_i$ is a vector. We will denote its dimension by $\|\hat{p}_i\|$.

If ρ_G is the graph spectrum, and $\hat{\rho}_1, \dots, \hat{\rho}_N$ is the estimated spectrum distribution of model M_i under parameter $\hat{p}_i$, then we can choose the optimum model M_i where

$$i = \arg\min_i 2KL(\rho_G, \hat{\rho}_i) + 2\|\hat{p}_i\|$$

Here, the factor $2\#(p_i)$ acts as a penalization term to avoid overfitting. Note that this term can be ignored if all parameters have equal dimensions.

This method allows us to extend our previously constructed parameter estimator to the selection of optimum models for a given graph. Computationally, we need to compute N estimators $\hat{p}_1, \dots, \hat{p}_N$, which can be computationally intensive. However, we can replace the KL function with any other distance measure between spectrum distributions (such as L_1 or L_2), allowing us to compute these estimators using a ternary search, significantly reducing the computation time.

3.3 ANOGVA

Given k groups of graph samples, can we establish whether or not they originate from the same graph population?

A perhaps naive approach is to select a (suitable) random graph model, estimate the parameter used for each graph, and use traditional ANOVA with the estimated parameters as input. However, for this to work, we first need to know which random graph model was used to generate the graphs, which is very unlikely in most realistic scenarios. Other non-parametric methods, like the Kolmogorov-Smirnov test, require independence of the graphs, which is often not true when they result from a biological process. Therefore, we will use an ANOVA-like approach following the ideas described by Fujita *et al* (André Fujita *et al.*, 2017) called ANOGVA.

In other words, we will perform a variation of the ANOVA using the complex distribution of eigenvalues of the graphs (Chavauty *et al.*, s.d.).

Suppose $g_1, \dots, g_k$ be k distinct graph populations. If these graphs come from the same population, their spectral distributions should be equal. Let ρ_{g_i} be the average graph spectrum for group i, $\rho_G = \frac{1}{k}\sum_{i=1}^{k} \rho_{g_i}$ be the overall graph spectrum average, and D be the Kullback-Leibler divergence.

The hypothesis that is being tested is the following:

$$H_0 \ : \ D(\rho_{g_1}, \rho_G) = D(\rho_{g_2}, \rho_G) = ... = D(\rho_{g_k}, \rho_G) = 0$$

$$H_1\text{: } D(\rho_{g_i}, \rho_G) \neq D(\rho_{g_j}, \rho_G) \text{ for some } i, j.$$

The alternative hypothesis is equivalent to stating that at least one of the graph populations was generated by a different process.

Under the null hypothesis, we expect the statistic $\Delta = \sum_{i=1}^{k} D(\rho_{g_1}, \rho_G)$ to be small. Under the alternative hypothesis, we expect it to be large.

The distribution of Δ is unknown and highly dependent on the used random graph model. Therefore, to test for significance, we will use a bootstrap approach.

The following algorithm describes how we compute the bootstrap.

```
Input: $k$ groups of graphs, $g_1, \ldots, g_k$, and a number of max-iterations $Max$
Output: A $p$-value
Estimate $\hat{\rho}_{g_1}$ and $\hat{\rho}_G$;
Calculate $\hat{\Delta} = \sum_{i=1}^{k} D(\hat{\rho}_{g_1}, \hat{\rho}_G)$;
Set $\hat{\Delta}_l = []$;
for $Max$ $iterations$ do
    Construct $k$ new groups $g_1', \ldots, g_k'$ by resampling (without replacement) the
      original graph set;
    Estimate the average spectrum distribution $\hat{\rho}_{g_i'}$ for each new graph $g_i'$;
    Calculate the overall graph spectrum average $\hat{\rho}_G'$;
    Calculate $\hat{\Delta}' = \sum_{i=1}^{k} D(\hat{\rho}_{g_1}', \hat{\rho}_G')$.;
    Append $\hat{\Delta}'$ to $\hat{\Delta}_l$;
end
Let $p$ = Cardinality($\hat{\Delta}' \in \hat{\Delta}_l$ : such that $\hat{\Delta}' \geq \hat{\Delta}$) $\cdot \frac{1}{Max}$;
return $p$;
```

Algorithm 1: ANOGVA

3.4 Permanogva

We are now interested in the following problem: Suppose we are given a population of graphs alongside a set of variables for each graph. Can we identify if the set of variables is related to the differences between the graph or if they are unrelated?

To verify this hypothesis, we will rely on a technique called permanova (Anderson, 2001; Anderson, 2017) (*permanogva* being the name given to its usage for graph spectra). This is a semi-parametric method in which we construct a pseudo-f statistic, a generalization of the standard F-statistic for classical statistics. Permanova tests for correlation between samples' inner distances and variables' inner distances (Zapala MA, 2006; Shehzad *et al.*, 2014). Given that we can use the graph spectrum to compare distances between graphs, we can test for correlation between graphs and a set of variables.

We start with a sample of k graphs and a set of t variables of interest for each graph.

Let $\mathbf{D}$ be a $k \times k$ distance matrix between each k graph in our sample. Let $\mathbf{X}$ be the $k \times t$ matrix of t variables of interest for each sample.

If $\mathbf{D}$ is written as $\mathbf{D} = \{d_{i,j}\}$, then let $\mathbf{A} = \{a_{i,j}\} = \{-\frac{1}{2}d_{i,j}^2\}$, and let $\mathbf{G}$ be Gower's centering matrix for $\mathbf{A}$, that is, if $\bar{a}_{i,.} = \frac{1}{k}\sum_{j=1}^{k} a_{i,j}$ is the average of line i in matrix $\mathbf{A}$, $\bar{a}_{.,j} = \frac{1}{k}\sum_{i=1}^{k} a_{i,j}$ is the average of column j in matrix $\mathbf{A}$, and $\bar{a}_{.,.} = \frac{1}{k^2}\sum_{i=1}^{k}\sum_{j=1}^{k} a_{i,j}$ is the overall average of matrix $\mathbf{A}$, then

$$\mathbf{G} = \{g_{i,j}\} = \{a_{i,j} - \bar{a}_{i,.} - \bar{a}_{.,j} + \bar{a}_{.,.}\}.$$

Now, inspired by the standard equations used in linear regression, let $\mathbf{H}$ be the hat matrix formed from $\mathbf{X}$

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t.$$

Now, we can finally construct the pseudo-F statistic. We define the among-group sum of squares as

$$SS_A = tr(\mathbf{HG}),$$

and, if $\mathbf{I}$ is the $k \times k$ identity matrix, we also define the residual sum of squares as

$$SS_R = tr((\mathbf{I} - \mathbf{H})\mathbf{G}).$$

Then, we define our pseudo-F statistic as

$$F = \frac{SS_A}{SS_R}\frac{k-t}{max(t-1,1)}.$$

We will perform a bootstrap method for the statistical test. We are testing for the following hypothesis:

H_0: There is no significance between the variables $\mathbf{X}$ and the distance between samples seen in the matrix $\mathbf{D}$

H_1: At least one of the variables in the matrix $\mathbf{X}$ is associated with the distance between samples seen in the matrix $\mathbf{D}$

Bootstrap is going to occur according to the following steps.

```
Input: A distance matrix $D$, A set of predictor variables $X$, and a number of
       max-iterations $Max$
Output: A $p$-value
Calculate an original pseudo-F statistic;
for $Max$ $iterations$ do
    Perform a sufficient amount of row-column permutations on the original
      distance matrix $\mathbf{D}$;
    Calculate the new pseudo-F statistic $F'$ using the original $\mathbf{H}$ matrix and the
      new $\mathbf{G}'$;
end
We calculate $p$ as the percentage of times $F'$ was at least as large as the original
  $F$;
return $p$;
```

Algorithm 2: Permanogva

3.5 K-Medoids

Suppose we have a set S of graphs, which came from k distinct populations of graphs, $S_1, \ldots, S_k$. We wish to assign each element of S to one of the groups.

Since we have well-defined notions of distance between graphs, we can use well-known techniques such as the k-medoids or the k-means methods to solve this problem by using the graph spectrum Hastie *et al.*, 2009.

Both methods were implemented to allow the usage of any of the distance functions described in section 2.3.

3.6 Implementation

The previous methods were implemented in R, extending the existing StatGraph package (Santos e A. Fujita, 2017). We constructed the multivariate kernel density estimator using the package 'ks' (Tarn Duong *et al.*, 2018) and optimized the parameter estimator using the package 'memoise' (Wickham *et al.*, 2021). The pre-release version of the statGraph codebase for directed graphs can be found in https://www.github.com/lesserfish/statGraph.

Capítulo 4

Simulations

To verify the power of the methods described in this paper, I constructed a set of simulations using the directed random graph models described in 2.4.

The following describes the simulations and their results.

4.1 Parameter estimation

I performed the following simulation to show the parameter estimator's power.

4.1.1 Simulation:

1. I generated a graph using a random graph model with the directionality technique described before.
2. The graph was created with parameters p_1, p_2 for specific values of $p_1, p_2 \in [0, 1]$
3. The search was done using the L_1 distance, with an epsilon of 0.01 using a ternary search.
4. This same experiment was repeated 1 000 times, creating a distribution of estimated parameters

I used the following random graph models: Erdős–Rényi (ER), Watts–Strogatz (WS), and the Barabási–Albert (PA). For the size of the graphs, I performed the simulation with $n = 100$, $n = 300$, $n = 500$, and $n = 800$.

In the first simulation, the parameters satisfied $p_1 = p_2$ so that the search could be done in one dimension. An additional simulation is included when $p_1 \neq p_2$, showcasing that the search can be done in several parameters simultaneously.

4.1.2 Results:

Figures 4.1 to 4.3 show the distribution of parameters for the first simulation (where $p_1 = p_2$) for all random graph models. Figure 4.4 shows the results of the second simulation (where $p_1 \neq p_2$).

We can see that increasing the size of the graph (and consequently increasing the number of eigenvalue points of its adjacency matrix) improves performance.

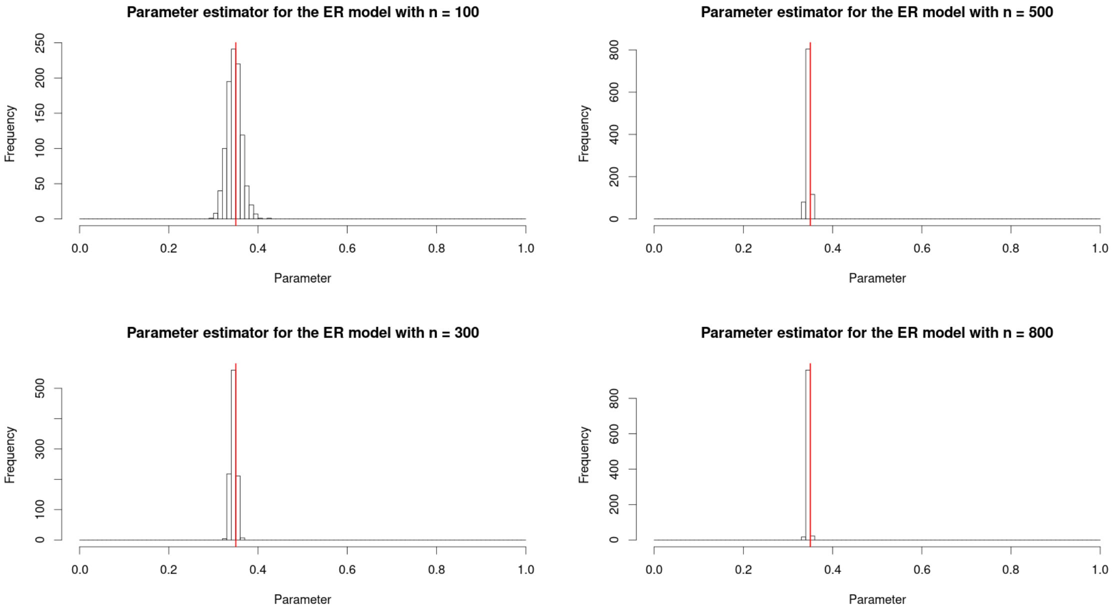


Figura 4.1: *Results of the parameter estimator simulation for the Erdős–Rényi model. A red vertical line highlights the correct parameter of* 0.35.

4.2 Model Selection

I performed the following simulation to show the power of the model selection.

4.2.1 Simulation:

1. I generated a graph with a given model
2. The graph was created with parameters $p_1, p_2 = 0.35$ and with sizes ranging from 20 to 120.
3. I performed the model selection using ER, PA, and WS models as candidates.
4. I used a ternary search with an epsilon of 0.01.
5. This same experiment was repeated 100 times for each size, allowing us to see how frequently each candidate model was chosen.

4.2.2 Results

Figure 4.5 shows the rate of selection of each candidate model as the size of the graph increases.

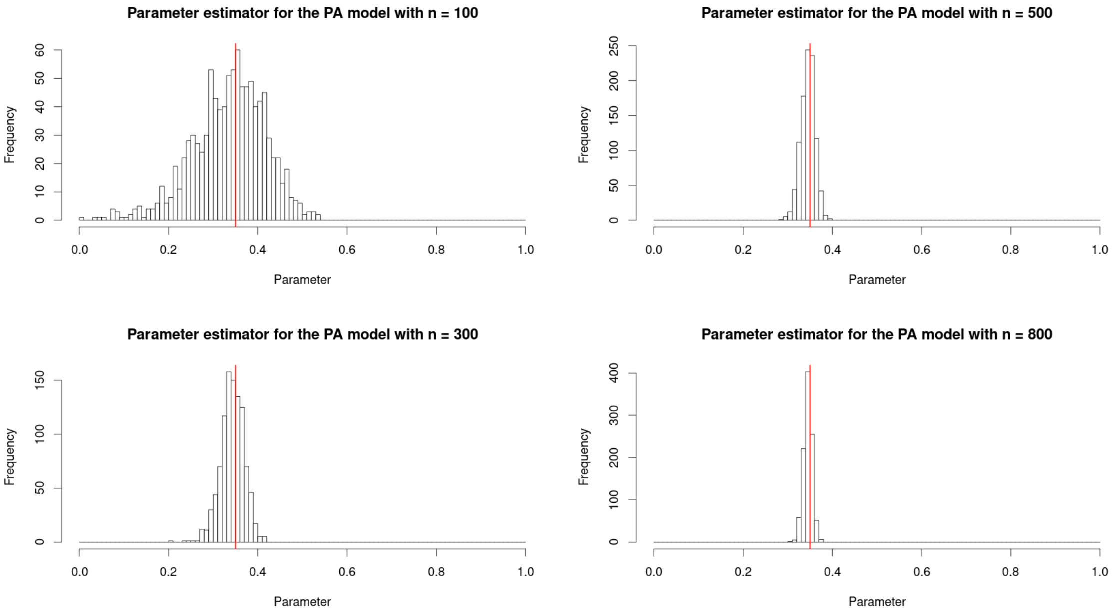


Figura 4.2: *Results of the parameter estimator simulation for the Barabási–Albert model. A red vertical line highlights the correct parameter of* 0.35.

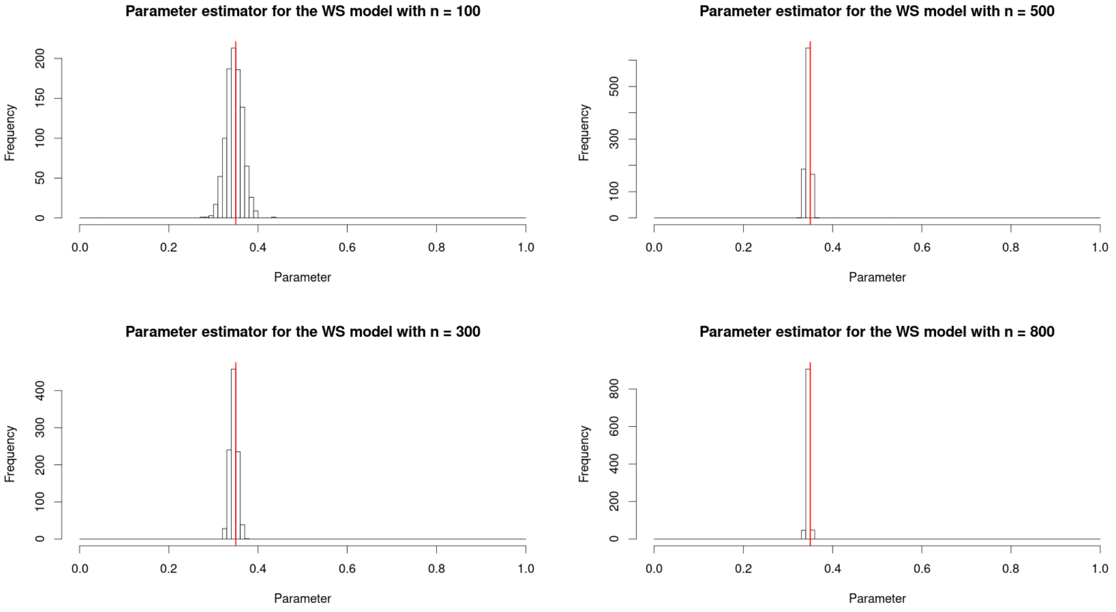


Figura 4.3: *Results of the parameter estimator simulation for the Watts–Strogatz model. A red vertical line highlights the correct parameter of* 0.35.

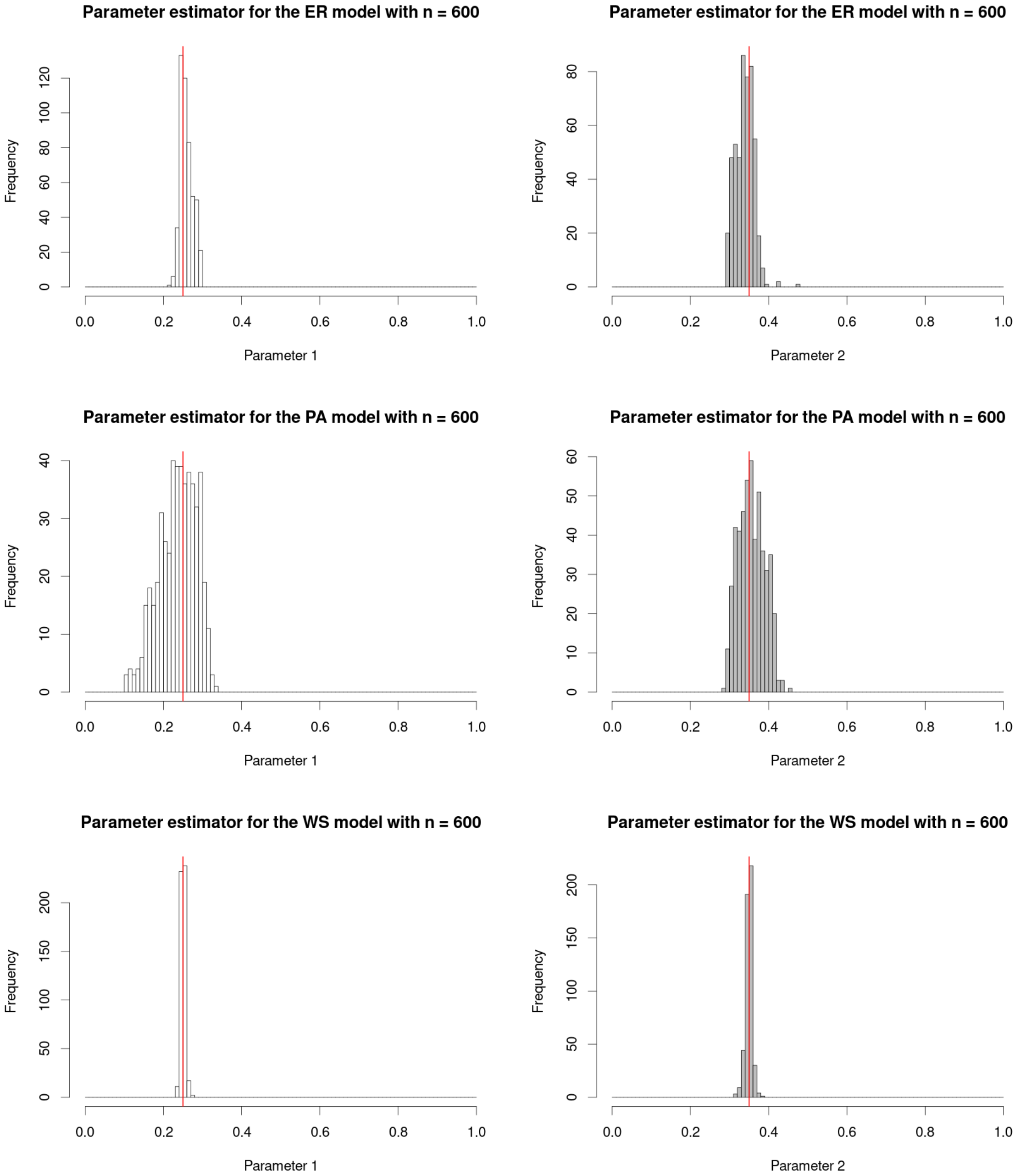
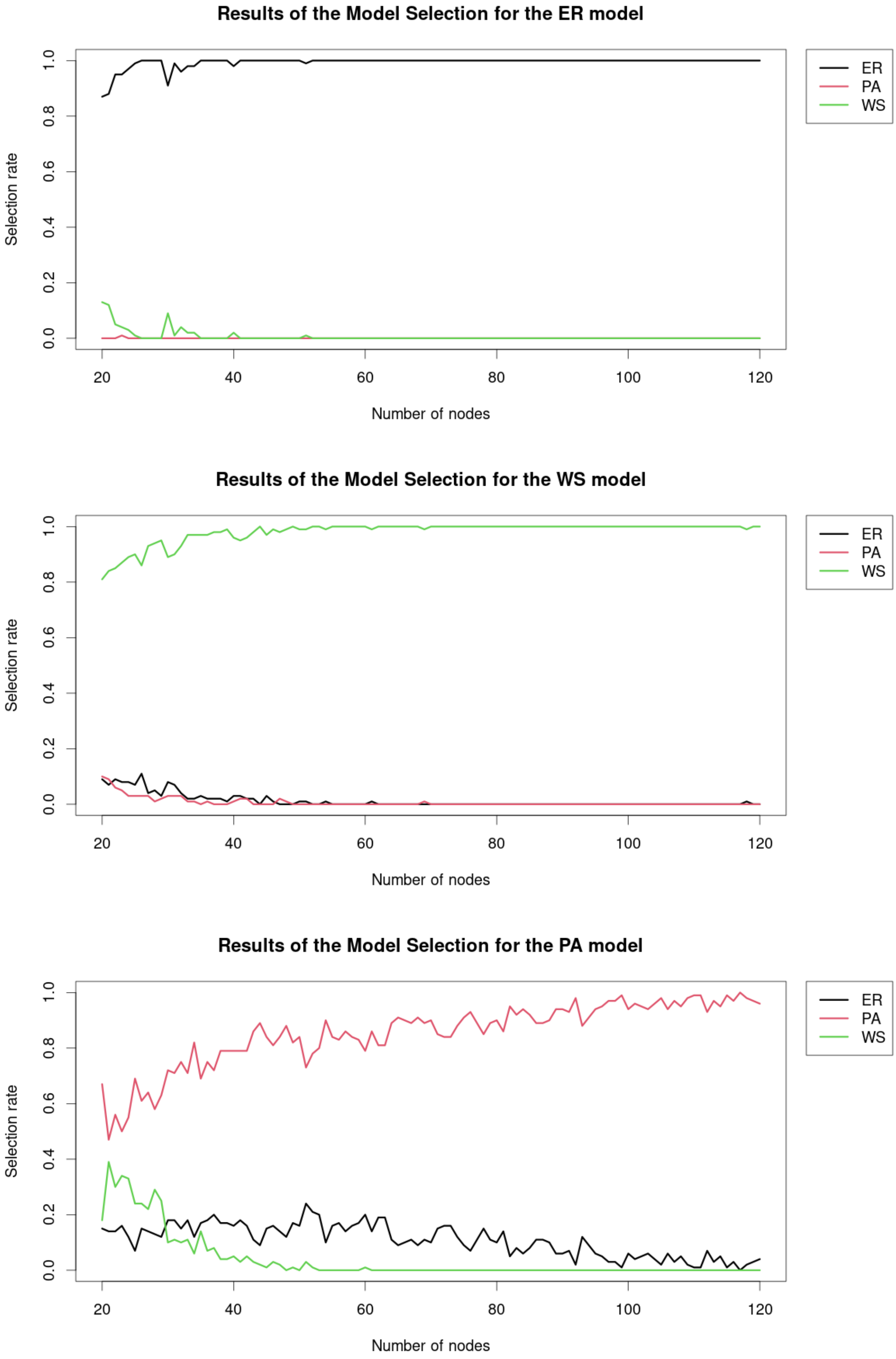


Figura 4.4: *Results of the 2-dimensional parameter estimator simulation for all models. On the left we can see the results for the first parameter, and on the right we can see the results for the second parameter. A red vertical line highlights the correct parameters of* $(0.25, 0.35)$.



Results of the Model Selection for the ER model

Results of the Model Selection for the WS model

Results of the Model Selection for the PA model

Figura 4.5: *Results of the Model Selection simulation for all models. The title indicates which model was used to generate the graph. On the y-axis, we see the rate of selection of each candidate model and on the x-axis, we see the number of nodes used to generate the graph.*

4.3 ANOGVA

To show the power of the ANOGVA method, I performed the following simulation.

4.3.1 Simulation:

1. I generated three groups of graphs. Each group contains 10 graphs with $n = 400$ nodes.
2. Groups 1 and 2 were generated with parameters $p_1 = p_2 = p$ for a specific value of $p \in [0, 1]$
3. Group 3 was generated using parameter $p_1 = p_2 = p + \epsilon$ for some small value of ϵ
4. We then performed the ANOGVA algorithm using 100 iterations of bootstrap, obtaining a p-value
5. This experiment was run 500 times, generating a distribution of p-values.

We expect that for very low values of ϵ, the p-values should follow mostly a uniform distribution. As ϵ increases, the p-values should be mostly close to 0.

The following random graph models were used: Erdős–Rényi (ER), Watts–Strogatz (WS), and the Barabási–Albert (PA).

4.3.2 Results:

We plot the results in an ROC plot. In this plot, the y-axis corresponds to the alpha-acceptance threshold ($\alpha = 1 - p$), and the x-axis corresponds to the acceptance rate. Figure 4.6 shows the results.

4.4 Permanogva

To show the power of the Permanogva method, I performed the following simulation.

4.4.1 Simulation:

1. I generated $N = 30$ samples of 2 variables V_1 and V_2, each variable V_i being distributed normally around 0 ($V_i \sim N(0, 1)$)
2. For each sample, I generated a parameter value of

   $$p = sigmoid(\frac{v_1}{2} - \frac{v_2}{2} + \epsilon)$$

   where ϵ is normally distributed around 0 and describes a random effect.
3. For each sample, I generated a graph using parameters $p_1 = p_2 = p$

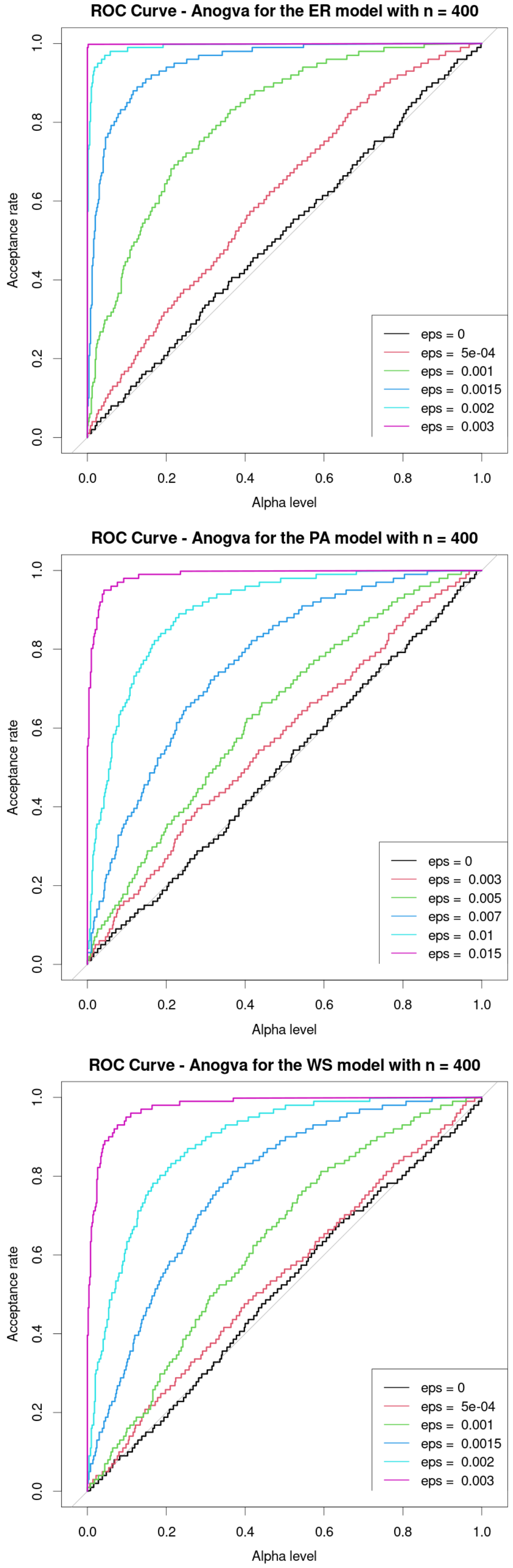


Figura 4.6: *Results of the ANOGVA simulation for all models for several group sizes. We can see that, as ϵ increases, ANOGVA quickly becomes capable of verifying that the groups are indeed different.*

4. A distance matrix of size 30 × 30 was computed, using the L_2 metric between the graph spectrum of each sample.
5. To simulate the alternative hypothesis, I calculated the permanogva result using this matrix $\mathbf{D}$ and using the predictor matrix $\mathbf{X} = [V_1, V_2]$.
6. To simulate the null hypothesis, I calculated the permanogva result using this matrix $\mathbf{D}$ and a random unrelated predictor matrix.
7. I repeated this same experiment 500 times, keeping track of the p-values.

4.4.2 Results:

Instead of showing the p-value distribution, we plot the results in an ROC plot. In this plot, the y-axis corresponds to the alpha-acceptance threshold, and the x-axis corresponds to the acceptance rate. Figure 4.7 shows the results.

4.5 K-medoids

To show the strength of K-medoids, I performed the following simulation.

4.5.1 Simulation:

I generated two groups of 20 graphs, G_1 and G_2. The second group was generated with the same parameter as the initial group, with an added epsilon. I used K-medoids using $k = 2$ with a distance matrix calculated using the L_1 distance. The ARI score (Hubert e Arabie, 1985) was used to establish performance. For each ϵ, the ARI score was calculated 30 times, and the mean and variances were used to create the plots.

4.5.2 Results:

Figures 4.8 to 4.10 show the ARI scores as we shift the ϵ from 0 to a higher number. We see that once we generate the graphs using a sufficiently different parameter, the spectral distance is enough for us to distinguish between populations of graphs efficiently.

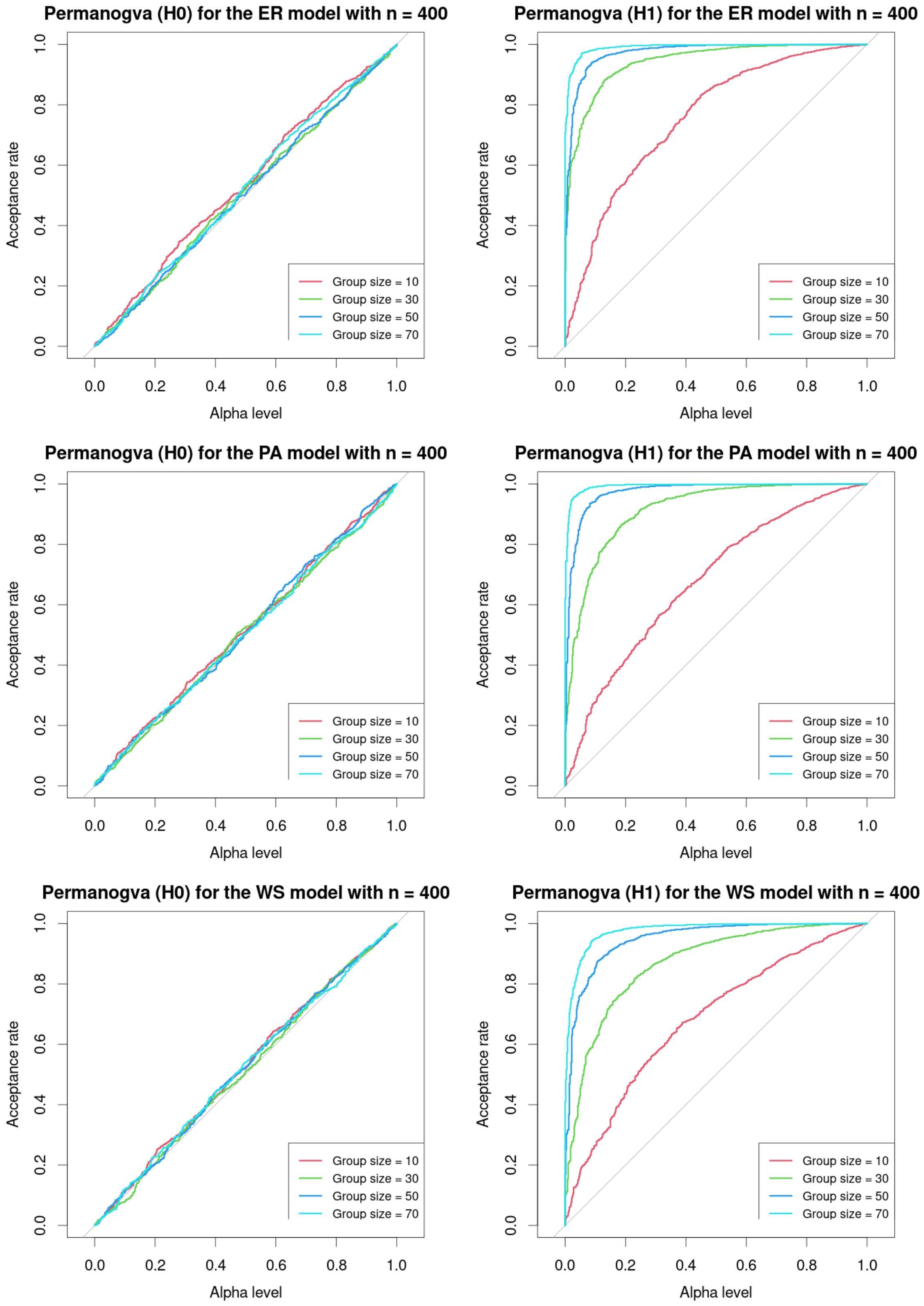


Figura 4.7: *Results of the Permanogva simulation for all models. On the left, we see the results when the conditions for the null hypothesis are satisfied, i.e., the graphs are generated independently from the predictor variables. On the right, we can see the results when conditions for the alternative hypothesis are satisfied, i.e., the graphs are generated with the predictor variables. We can see that when the group size increases, the power of Permanogva significantly increases.*

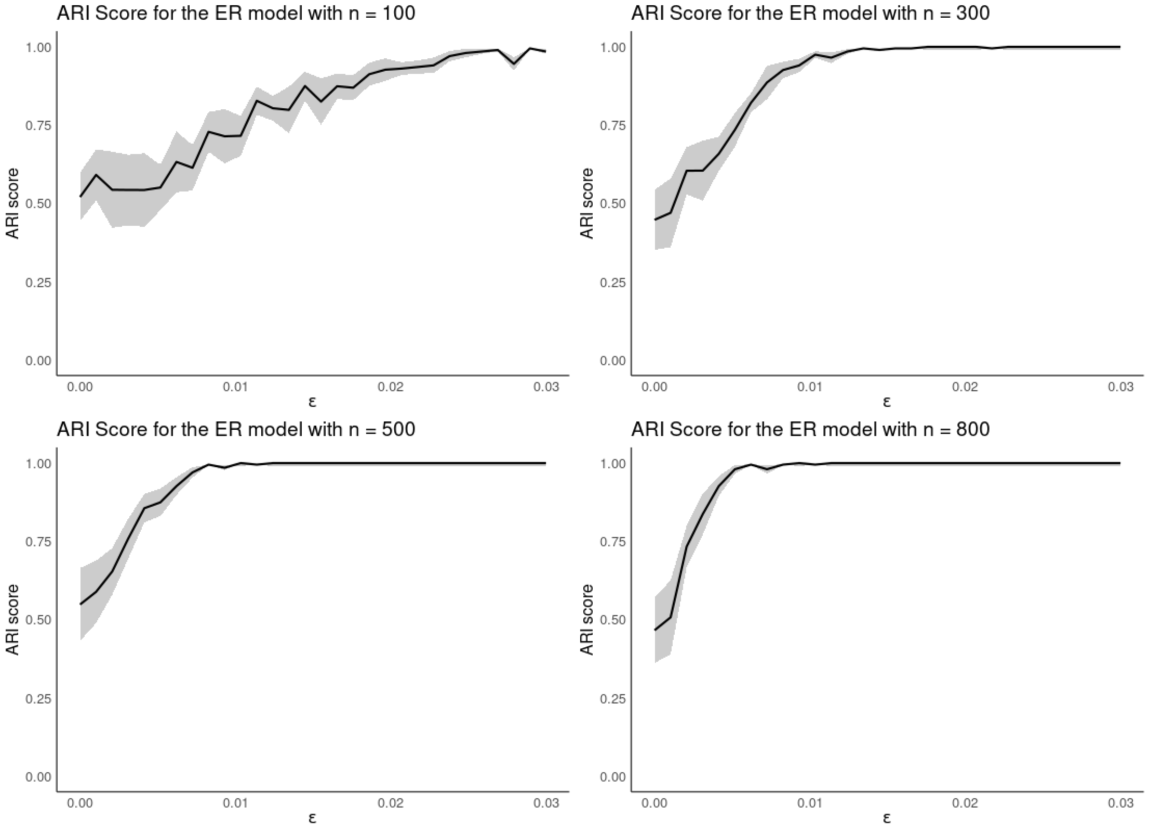


Figura 4.8: *Results of the K-Medoids simulation for the Erdős–Rényi model. The line represents the mean of the ARI score. The shaded area indicates the* 95% *confidence interval. As we increase the value of* ϵ, *the method rapidly can correctly cluster the graph populations. We also note that as we increase the size of the graphs, the method becomes stronger.*

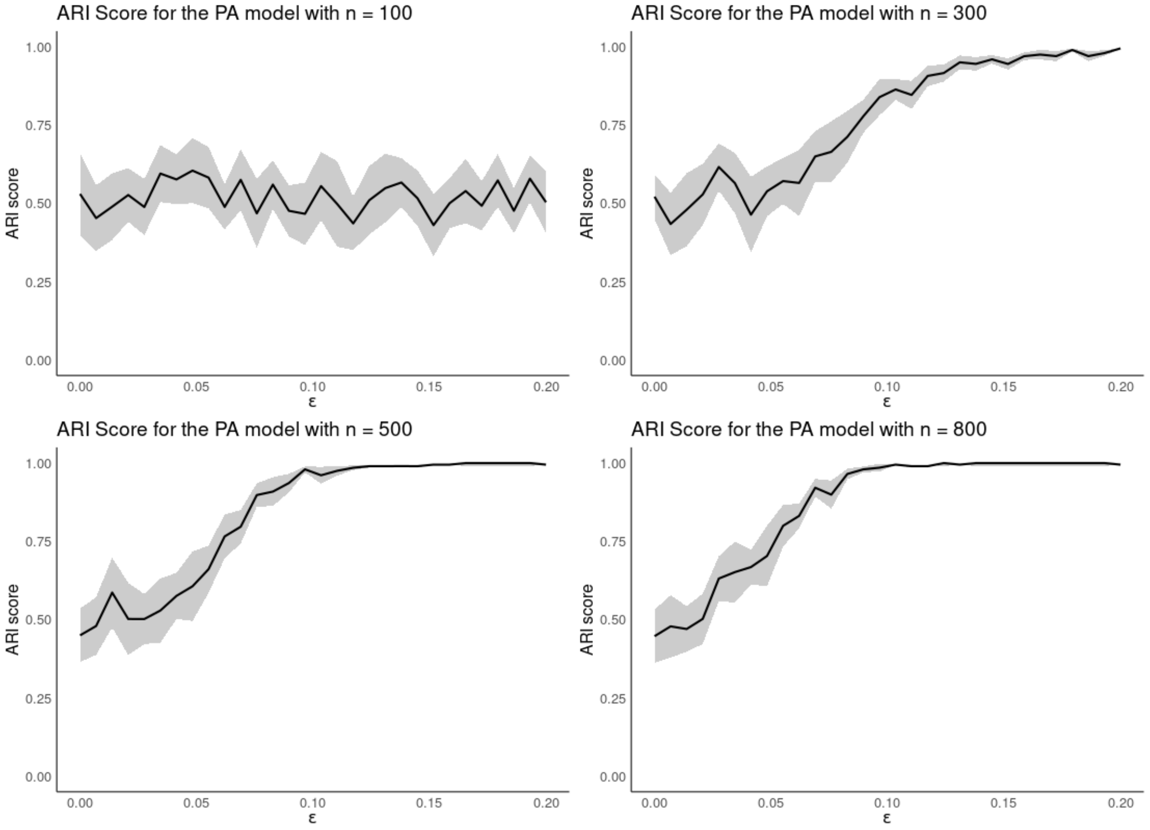


Figura 4.9: *Results of the K-Medoids simulation for the Barabási–Albert model. The line represents the mean of the ARI score. The shaded area indicates the* 95% *confidence interval. As we increase the value of* ϵ*, the method rapidly can correctly cluster the graph populations. We also note that as we increase the size of the graphs, the method becomes stronger.*

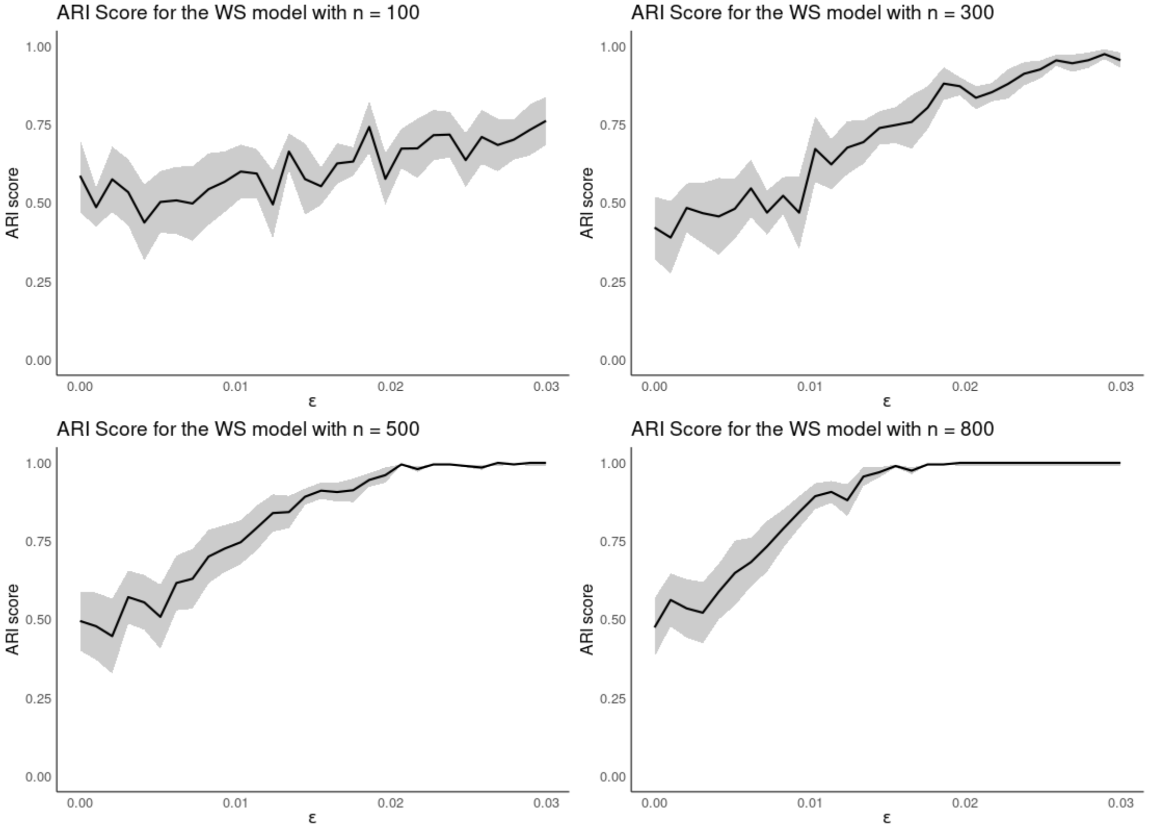


Figura 4.10: *Results of the K-Medoids simulation for the Watts–Strogatz model. The line represents the mean of the ARI score. The shaded area indicates the* 95% *confidence interval. As we increase the value of* ϵ*, the method rapidly can correctly cluster the graph populations. We also note that as we increase the size of the graphs, the method becomes stronger.*

Capítulo 5

Applications to Biological data

To show the efficacy of the techniques described in this article, I used the ANOGVA method (Chavauty *et al.*, s.d.) on a biological dataset consisting of 5 groups of graphs obtained from one of the experiments associated with Project Tycho, whose goals are to share reliable massive neural and behavioral data for understanding brain mechanism.

The purpose of the particular experiment was to compare neural activity between most of the lateral cortex measured with electrocorticographic signals (ECoG) in a macaque during five stages: awake with eyes opened, Awake with eyes closed, Anesthetized, Recovering with eyes closed, and recovering with eyes open.

5.1 Data Source

The data source used is titled 'Anesthesia and Sleep Task' and was obtained from Project Tycho and downloaded via their website at wiki.neurotycho.org/.

Four experiments were conducted on a different monkey (Yanagawa *et al.*, 2013). In each experiment, a monkey was seated in a primate chair with restricted arms and head movement. In particular, for the monkey named George, the following steps describe the experiment:

Neural data was acquired through 128 ECoG electrodes measuring ECoG signals from most of the lateral cortex. Neural activity was recorded during all of the following stages. Initially, the monkey was awake and opened its eyes, sitting calmly in his chair for 10 minutes. Next, the eyes of the monkey were covered with an eye mask to avoid evoking a visual response. The monkey was left sitting in his chair for another 10 minutes. Recording of neural activity was stopped while anesthesia was intramuscularly injected into the monkey. By the point at which the monkey had stopped responding to manipulation of the monkey's hand or touching the nostril or philtrum with a cotton swab, neural activity recording was resumed for another 20 minutes. After the anesthetized condition, the monkey recovered from the anesthesia and was left alone for 55 minutes with its eyes still covered. Next, the eye mask was removed, and the monkey was left to sit calmly on his chair for another 10 minutes.

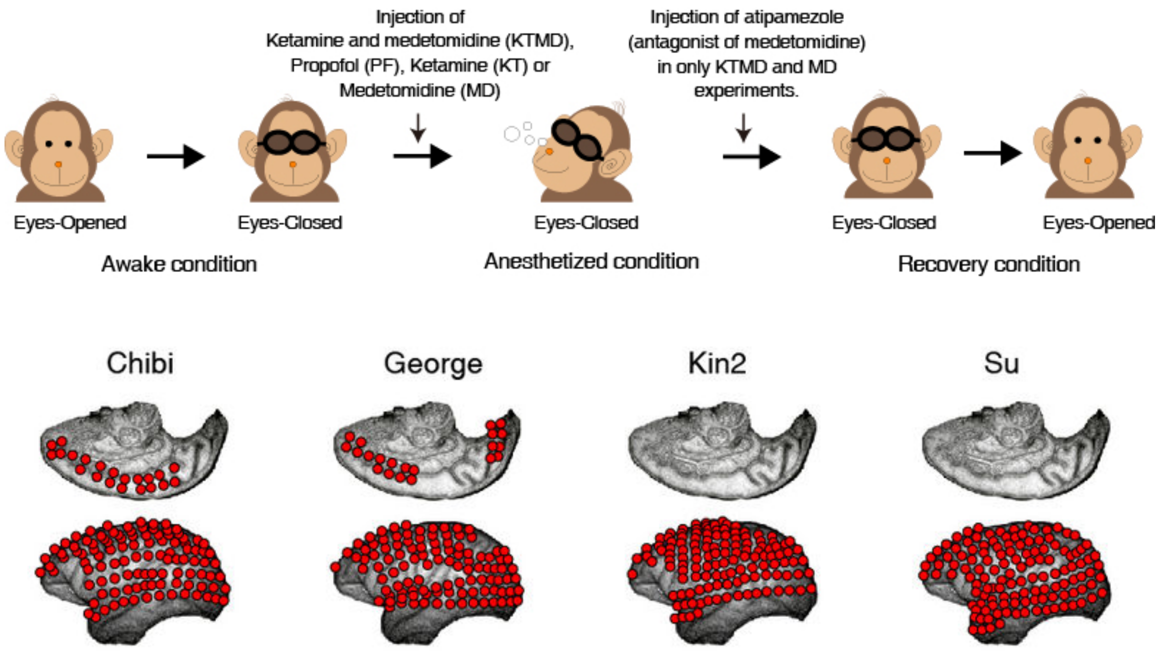


Figura 5.1: *Description of the Anesthesia and sleep task* YANAGAWA *et al., 2013.*

5.2 Data processing and graph generation

The initial data generated by the experiment consisted of 128-time series in 5 categories: conscious with open eyes, conscious with closed eyes, anesthetized, recovering with closed eyes, and recovering with open eyes (CHAVAUTY *et al.*, s.d.).

Initially, the data was processed through several finite impulse response (FIR) filters to remove any effect caused by electrical interference (in particular, interference caused by electrical sockets).

The filtered data was divided into several time windows, each lasting 4 seconds. The graphs were generated using the generalized partial directed coherence (gPDC) (SAMESHIMA e BACCALA, 2016).

The gPDC is a frequency domain approach to identify the direction of information flow (Granger causality) between multiple time series. We say that a time series X Granger causes another time series Y if knowledge of $X(t-1), \dots, X(t-k)$ increases the prediction of $Y(t)$.

We carried out gPDC on the 128 frequencies of the filtered data. The result was five sets of 128 groups of graphs (one for each generated frequency). Each group consisted of several graphs, each representing a time window in its category. Each graph had 128 nodes (each corresponding to a different ECoG electrode). The graph was directed and weighted, where each edge between two nodes corresponded to the level of causality between the ECoG electrodes.

5.3 ANOGVA

To verify the power of the ANOGVA method, I performed the following experiment. I selected a single-frequency domain on which the graphs were to be obtained. Given that frequency, I chose 100 graphs from each of the 5 categories. This left me with:

1. G_1: 100 graphs generated from the period when the monkey was awake with its eyes opened
2. G_2: 100 graphs generated from the period when the monkey was awake with its eyes closed
3. G_3: 100 graphs generated from the period where the monkey was anesthetized
4. G_4: 100 graphs generated from the period when the monkey was recovering with its eyes closed
5. G_5: 100 graphs generated from the period when the monkey was recovering with its eyes closed

Each of these graphs measured the flow of information between distinct brain segments.

5.3.1 First Experiment:

I first performed an ANOGVA test using the 5 groups. I was testing the following hypothesis.

H_0: There is no difference between the information flow in any of the 5 categories versus

H_1: There is a difference between the information flow in at least two of the categories

This experiment was run with a 1 000 iterations bootstrap.

5.3.2 Second Experiment:

I then performed the same experiment but using pairwise groups. In specific, for every two distinct groups G_i and G_j, I performed ANOGVA using only these two groups as an input, thus testing the following hypothesis:

H_0: There is no difference between the information flow between G_i and G_j

H_1: There is a difference between the information flow between G_i and G_j

This experiment was also run with a bootstrap of 1000 iterations.

5.3.3 Third Experiment:

Since all graphs originate from the same monkey, there is a possibility that obtaining low p-values in the previous experiments is not a consequence of the difference between the distinct categories. To verify if the significance of the previous experiments was valid, I performed an ANOGVA test between each group and itself. In specific, I performed the following for each group G_i

1. I split group G_i into two randomly sampled groups with no replacement, obtaining $G_{i,1}$ and $G_{i,2}$
2. I performed an ANOGVA test on these groups with a bootstrap of 300 iterations.
3. I stored the calculated p-value.
4. I repeated this 300 times, generating a distribution of p-values.

If low p-values are explained by the fact that all graphs originate from the same monkey, then performing ANOGVA using the setup described above should give us mostly low p-values.

5.4 Results

5.4.1 First Experiment:

For the first experiment, we obtained a p-value less than the maximum sensitivity of the bootstrap method of 0.001. This shows that there is indeed a significant difference between information flow between at least two of the groups.

5.4.2 Second Experiment and Third experiment:

The results of the second and third experiments also indicate a strong difference between each population group. Table 5.1 shows the p-values obtained when comparing groups G_i and G_j with the ANOGVA method, with 1 000 iterations of the bootstrap. In figure 5.2, we see five images, which are the distribution of obtained p-values when comparing each group with itself.

	G_1	G_2	G_3	G_4	G_5
G_1		0.002	< 0.001	< 0.001	< 0.001
G_2			< 0.001	< 0.001	0.076
G_3				< 0.001	< 0.001
G_4					< 0.001
G_5					

Tabela 5.1: *Results of second experiment. We see the p-value obtained when comparing each group with each other, using the ANOGVA method with a maximum sensitivity of* 0.001.

We can see that we obtain low p-values when comparing distinct groups. Any fear that this might be because both groups originate from the same monkey can be eased by looking at the results of the third experiment. We note a well-defined uniform distribution in each experiment, proving that the graphs from the same monkey are insufficient to justify a low p-value between groups.

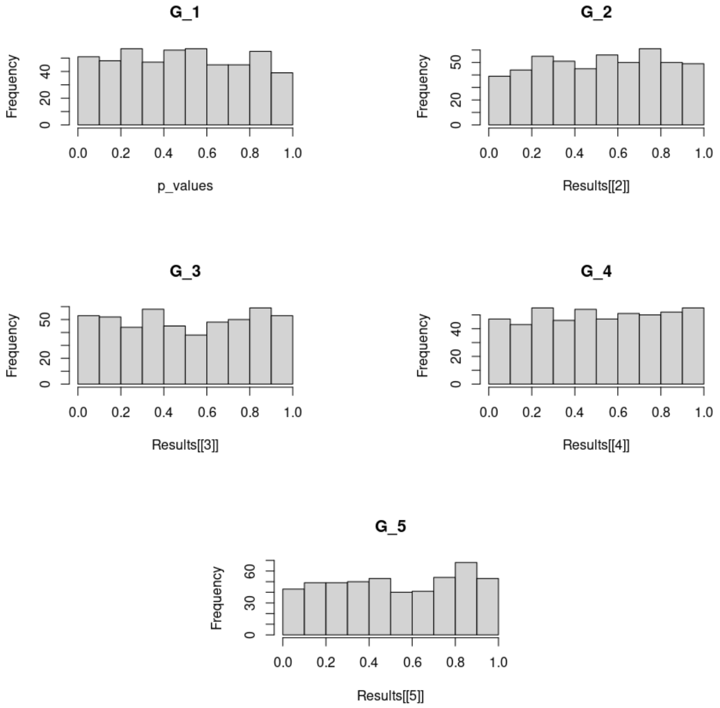


Figura 5.2: *Results of the third experiment. We can see the distribution of p-values when comparing each group with itself using the ANOGVA method.*

We note that we could effectively distinguish between the groups, demonstrating our methods' efficiency against biological data.

Capítulo 6

Conclusion

We have developed several statistical techniques to

1. Estimate the parameter used to create a directed graph
2. Establish which random graph model was used to create a directed graph
3. Establish whether two or more populations of graphs differ
4. Verify correlation between a group of graphs and predictor variables
5. Cluster a group of graphs

We showed that we could also use our techniques to effectively distinguish between the brain interaction network of a monkey under different phases of anesthesia. I believe that more profound studies in computational methods related to graph spectra may help us highlight more accurately the differences in the topology of these networks and identify common graph topology patterns under different brain states.

There is already essential work being done in the area. In 1977, Vidal coined the term brain-computer interface (BCI) and investigated applications of real-time detection of brain events in EEG (Vidal, 1977). More recently, Neuralink's company implanted its first brain chip in a human (Hern, 2024). Like the Project Tycho experiment, Neuralink uses time series data of electrical brain activity as input. There is significant interest in recognizing common brain patterns, both in the medical and private Neurotechnology fields. The approaches described here and in the original article by Takahashi (Takahashi *et al.*, 2012) are mathematically flexible to allow their implementation in future technologies.

Referências

[Alon 2006] Uri Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits.* Chapman & Hall/CRC Mathematical e Computational Biology, 2006 (citado na pg. 1).

[Anderson 2001] Marti J. Anderson. "A new method for non-parametric multivariate analysis of variance". Em: *Austral Ecology* 26 (2001), pgs. 32–46 (citado na pg. 12).

[Anderson 2017] Marti J. Anderson. *Permutational Multivariate Analysis of Variance (PERMANOVA).* John Wiley & Sons, Ltd, 2017, pgs. 1–15. isbn: 9781118445112. doi: https://doi.org/10.1002/9781118445112.stat07841 (citado na pg. 12).

[Barabasi e Oltvai 2004] A.-L. Barabasi e Z. N. Oltvai. "Network biology: understanding the cell's functional organization". Em: *Nat. Rev. Genet.* 5 (2004), pgs. 101–113 (citado na pg. 1).

[Barabási e Albert 1999] A. L. Barabási e R. Albert. "Emergence of scaling in random networks". Em: *Science* 286 (1999), pgs. 509–512 (citado na pg. 6).

[Bullmore e Sporns 2009] E. Bullmore e O. Sporns. "Complex brain networks: graph theoretical analysis of structural and functional systems". Em: *Nat. Rev. Neurosci.* 10 (2009), pgs. 186–198 (citado na pg. 1).

[Chavauty *et al.* s.d.] Victor Chavauty, Eduardo Lira e André Fujita. "Spectrum-based statistical methods for directed graphs with applications in biological data". Em: *Advances in Bioinformatics and Computational Biology, Lecture Notes in Computer Science* 13954 () (citado nas pgs. 2, 4, 11, 27, 28).

[Cover e Thomas 2006] Thomas M Cover e Joy A Thomas. *Elements of Information Theory.* 2nd. New Jersey: Wiley-Interscience, 2006 (citado na pg. 5).

[Daniel Yasumasa Takahashi *et al.* 2017] Joana Bisol Balardin Daniel Yasumasa Takahashi André Fujita, Maciel Calebe Vidal e João Ricardo Sato. "Correlation between graphs with an application to brain network analysis". Em: *Computational Statistics & Data Analysis* 109 (2017), pgs. 76–92 (citado na pg. 2).

[Tarn Duong *et al.* 2018] Tarn Duong, Matt Wand, Jose Chacon e Artur Gramacki. *ks: Kernel Smoothing*. https://CRAN.R-project.org/package=ks. 2018 (citado na pg. 14).

[Tran Duong 2007] Tran Duong. "ks: kernel density estimation and kernel discriminant analysis for multivariate data in r". Em: *Journal of Statistical Software* 21.7 (2007), pgs. 1–16 (citado na pg. 4).

[Erdős e Rényi 1959] P. Erdős e A. Rényi. "On random graphs i". Em: *Publ. Math. Debrecen* 6 (1959), pgs. 290–297 (citado na pg. 6).

[Andre Fujita *et al.* 2020] Andre Fujita, Eduardo Silva Lira e Suzana de Siqueira Santos. "A semi-parametric statistical test to compare complex networks". Em: *Journal of Complex Networks* 8 (2020) (citado nas pgs. 1, 4).

[André Fujita *et al.* 2017] André Fujita, Maciel C. Vidal e Daniel Y. Takahashi. "A statistical method to distinguish functional brain networks". Em: *Frontiers in Neuroscience* 11 (2017) (citado na pg. 11).

[Hastie *et al.* 2009] T. Hastie, R. Tibshirani e J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2009. isbn: 9780387848846. url: https://books.google.com.br/books?id=eBSgoAEACAAJ (citado na pg. 14).

[Hern 2024] Alex Hern. *Elon Musk says Neuralink has implanted its first brain chip in human*. Article retrieved on 2024-01-30. The Guardian. Jan. de 2024. url: https://www.theguardian.com/technology/2024/jan/29/elon-musk-neuralink-first-human-brain-chip-implant?CMP=share_btn_url (citado na pg. 33).

[Hubert e Arabie 1985] L. Hubert e P. Arabie. "Comparing partitions." Em: *Journal of Classification 2* (1985) (citado na pg. 22).

[MacKay 2003] David J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. First. Cambridge University Press, 2003, pg. 34. isbn: 9780521642989 (citado na pg. 6).

[Ramos *et al.* 2023] Taiane Coelho Ramos, Janaina Mourão-Miranda e André Fujita. "Spectral density-based clustering algorithms for complex networks". Em: *Frontiers in Neuroscience* 17 (2023) (citado na pg. 2).

[Ribeiro *et al.* 2021] A.H. Ribeiro, M.C. Vidal, J.R. Sato e A. Fujita. "Granger causality among graphs and application to functional brain connectivity in autism spectrum disorder". Em: *Entropy* 23 (2021), pg. 1204 (citado na pg. 2).

[Sameshima e Baccala 2016] Koichi Sameshima e Luiz Baccala. *Methods in brain connectivity inference through multivariate time series analysis*. Abr. de 2016, pgs. 1–251 (citado na pg. 28).

[Santos e A. Fujita 2017] S. S. Santos e A. Fujita. *statGraph: Statistical Methods for Graphs*. https://cran.r-project.org/package=statGraph. 2017 (citado na pg. 14).

[Scott 2012] J. Scott. *Social Network Analysis*. Sage, 2012 (citado na pg. 1).

[Shehzad *et al.* 2014] Z. Shehzad *et al.* "A multivariate distance-based analytic framework for connectome-wide association studies". Em: *Neuroimage* 93 (2014) (citado na pg. 12).

[Siqueira Santos *et al.* 2014] S. de Siqueira Santos, D. Y. Takahashi, A. Nakata e A. Fujita. "A comparative study of statistical methods used to identify dependencies between gene expression signals." Em: *Brief. Bioinform.* 15 (2014), pgs. 906–918 (citado na pg. 1).

[Takahashi *et al.* 2012] Daniel Yasumasa Takahashi, João Ricardo Sato, Carlos Eduardo Ferreira e André Fujita. "Discriminating different classes of biological networks by analyzing the graphs spectra distribution". Em: *PLOS ONE* 7.12 (dez. de 2012), pgs. 1–12. doi: 10.1371/journal.pone.0049949. url: https://doi.org/10.1371/journal.pone.0049949 (citado nas pgs. 1, 5, 11, 33).

[Van Mieghem 2011] Piet Van Mieghem. *Graph Spectra for Complex Networks*. Cambridge University Press, 2011 (citado na pg. 5).

[Vidal 1977] J Vidal. "Real-time detection of brain events in eeg". Em: *Proceedings of the IEEE* 65.5 (1977), pgs. 633–641. doi: 10.1109/PROC.1977.10542 (citado na pg. 33).

[Watts e Strogatz 1998] D. Watts e S. Strogatz. "Collective dynamics of 'small-world' networks". Em: *Nature* 393 (1998), pgs. 440–442 (citado na pg. 6).

[Wickham *et al.* 2021] Hadley Wickham *et al. memoise: 'Memoisation' of Functions*. https://CRAN.R-project.org/package=memoise. 2021 (citado na pg. 14).

[Yanagawa *et al.* 2013] T. Yanagawa, Z. C. Chao, N. Hasegawa e N. Fujii. "Large-scale information flow in conscious and unconscious states: an ecog study in monkeys". Em: *PloS one* 8 (2013) (citado nas pgs. 27, 28).

[Zapala MA 2006] Schork NJ Zapala MA. "Multivariate regression analysis of distance matrices for testing associations between gene expression patterns and related variables". Em: *Proceedings of the National Academy of Sciences of the United States of America* 103 (2006) (citado na pg. 12).