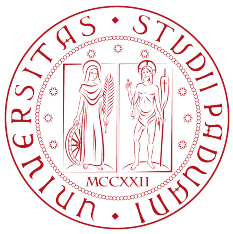

Università degli Studi di Padova

Department of Physics and Astronomy "Galileo Galilei"

Master Degree in Physics of Data

Final Dissertation

Generalized Thermodynamics in Complex Information Dynamics: Optimization Techniques and Applications to Mammalian Connectomes

Supervisor
Prof. Manlio de Domenico

Candidate
Mojtaba Roshana

Academic Year 2023/2024

Abstract

Natural and artificial complex systems, composed of numerous interacting components like neurons in the brain or individuals in societies, exhibit intricate structures with varying degrees of complexity. Understanding the principles governing these networks is a fundamental question in physics, as it enables the identification of networks shaped by similar underlying rules. Recent studies on mammalian connectomes' topological features, such as average shortest path length and clustering coefficient, reveal notable intra-order similarities with the phylogenetic tree taxonomy. However, a comprehensive understanding of brain functionality across different species requires examination not only of structural pathways but also of the flow of information—i.e., electrochemical signals—through them.

Utilizing recent advances in statistical physics, this research employs network density matrices to characterize and compare the short- to long-range information flow in mammalian connectomes. Our findings confirm high intra-order similarity along six orders of mammalians. By employing an unsupervised learning pipeline, we cluster mammalian connectomes based on their density matrix proximity, and entropy-free energy profile across various scales, uncovering some similarities to phylogenetic relationships. Furthermore, the results from the functional analysis are consistent with those from the topological analysis, reinforcing the dependency of connectome taxonomy on phylogenetic trees and highlighting the mutual information embedded in density matrices across different scales.

To gain deeper insights into real networks, we optimize procedures for aligning theoretical models with real-world biological networks, particularly the mammalian connectome. We quantify network distances using entropy profiles derived from the density matrix formalism and apply optimization algorithms such as Simulated Annealing and Particle Swarm Optimization. These methods minimize the distance between synthetic and empirical networks, effectively capturing information pathways across various scales. By exploring different parameters of generative models, we generate networks that closely resemble the target network. We conducted a series of experiments on synthetic networks to compare different generative models. These experiments focused on evaluating and analyzing the performance of each model in replicating the functional properties of the networks. The comparison was carried out using metrics like Jensen-Shannon Divergence (JSD) to assess how well each model captured the characteristics of the target network. Finally, we validate our approach by fitting synthetic networks to a real mammalian connectome from the MAMI dataset, demonstrating the effectiveness of our optimized generative models in capturing the intricate details of biological networks. This work advances the understanding of connectome structures and information flow while providing robust optimization techniques for modeling complex biological systems.

Acknowledgements

My deep interest in understanding the statistical physics governing complex systems led me to Prof. Manlio De Domenico, who generously shared his insights and provided invaluable guidance on how to advance his innovative theories on the statistical physics of complex networks. Since February 2023, when I joined the CoMuNe (Complex Multilayer Networks) Lab under his supervision, I have had the privilege of collaborating closely with him on my thesis. Prof. De Domenico's expertise and vision have been central to shaping my research, and his mentorship has profoundly influenced my academic growth.

I would also like to extend my gratitude to Dr. Arsham Ghavasieh, whose encouragement and assistance were invaluable in advancing this work. His support throughout my research was instrumental and greatly appreciated.

Contents

List of Figures

List of Tables

Chapter 1

Complex Networks

In our everyday world, many systems are made up of interacting parts, like neurons in the brain. Understanding these complex systems is essential because they play a crucial role in various natural phenomena. By using graph theory, we can represent these systems as networks, and with the help of statistical mechanics, we can better describe and analyze how they behave. In this chapter, we lay out the basics of complexity science and show how networks can be a powerful way to explain complex systems.

The first part of this chapter begins by defining complex systems. It highlights the importance of network structure in determining how these systems function as a whole. We explore how network dynamics influence system properties and discuss the philosophical ideas behind this approach. Additionally, we provide an overview of the mathematical tools needed to model the network structure of complex systems. This sets the foundation for deeper analysis in the sections that follow. Next, we introduce generative models to enhance the understanding of network models. Finally, we define key topological features that help explain the properties of networks.

1.1 Complex Systems and Networks

Scientists have always observed groups of similar units showing intelligent behavior. These include the coordinated movements of birds flying together or the synchronized electrical activity in large networks of neurons in the human brain. However, traditional approaches that focus on studying these units individually often struggle to explain the properties that emerge when these units come together. For instance, even with a deep understanding of a single neuron's physiology, it's not enough to fully grasp higher-level brain functions like consciousness and memory. These complex properties arise from the collective dynamics of billions of interacting neurons, suggesting that it's the interactions between these units that lead to complex behaviors.

In complex systems, interactions between units are not random; instead, they are organized into intricate structures that govern how information is exchanged. For example, in parts of the brain where neurons are closely packed, their closeness allows for quick information flow, while in less connected areas, the structure may slow down communication. This underlines the crucial role of interaction networks in enabling complex behavior. The architecture of these interaction networks

greatly influences a system's overall function. This impact is evident in the survival mechanisms of living organisms, the efficiency of city transportation systems, and the emergence of the human brain. However, a fundamental question in complexity science remains: how do structure and function interact, and which is more crucial to our understanding of complex networks?

The core philosophical question of this thesis is centered around understanding the information processes that underlie complex interactions. Can we use the flow of information to identify similarities between different complex networks? This research uses concepts from statistical physics to explore these similarities through functional analysis, with a focus on the diffusion mechanism to examine information flow at different scales, particularly using a tool called the density matrix to study the connectomes. The properties that emerge in complex systems are closely tied to the interactions among their units. These interactions are shaped by the underlying structure. Physicists have developed mathematical frameworks based on networks to capture these properties. By modeling interactions, such as continuous and discrete diffusion, they have made significant progress in understanding and explaining complexity.

1.2 Network Modeling and Graph Theory

Network modeling is a powerful method for understanding and predicting the behavior of complex systems. In this approach, components are represented as nodes, and their interactions are depicted as edges. This representation facilitates the analysis of the underlying structure and dynamics of various systems. Graph theory provides the framework for modeling systems as networks. While network modeling is a broad field with many complexities, understanding some basic principles is essential for effectively modeling systems as networks.

A graph is a structure consisting of a set of points, generally called nodes or vertices, and lines connecting these, variously called edges or links. Graphs can represent many kinds of real systems, including social networks, communication networks, and biological systems such as neural networks. This section will outline the basic principles behind the modeling of networks, focusing especially on how the structure of networks can be represented using the mathematics of graph theory.

1.2.1 Adjacency Matrix

The adjacency matrix represents the structure of a graph mathematically. For a graph with N nodes, the adjacency matrix $\mathbf{A}$ is an $N \times N$ square matrix where each element A_{ij} indicates whether there is an edge between nodes i and j:

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge from } i \text{ to } j, \\ 0 & \text{otherwise.} \end{cases} \tag{1.1}$$

For undirected graphs, the adjacency matrix is symmetric ($A_{ij} = A_{ji}$), while for directed graphs, it is not necessarily symmetric. In a directed graph, there is a specific direction for the links. This means that a link from node i to node j does not imply a link from j to i. Figure 1.1 indicates an example of a directed graph.

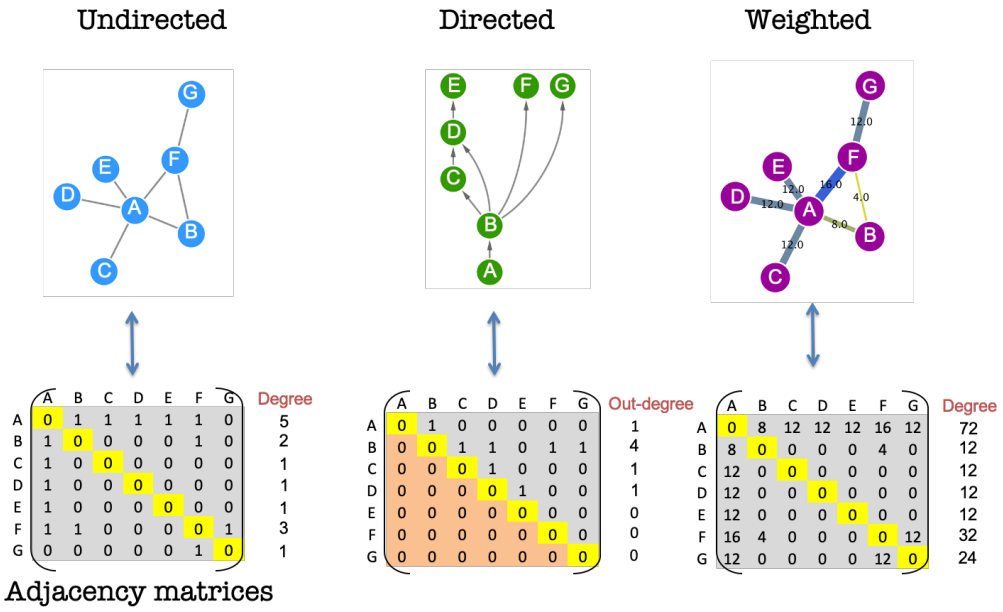

Figure 1.1: Example of network types and their adjacency matrices. The left shows an undirected network with a symmetric matrix. The center depicts a directed network with an out-degree matrix. The right illustrates a weighted network, where the matrix includes edge weights. Figure from Ref. [20].

In weighted graphs, the adjacency matrix is generalized to include weights:

$$W_{ij} = \begin{cases} \text{weight of the edge from } i \text{ to } j & \text{if there is an edge,} \\ 0 & \text{if there is no edge.} \end{cases} \tag{1.2}$$

The adjacency matrix for binary graphs is a special case of the weighted adjacency matrix where all non-zero entries are 1. Figure 1.1 presents examples of weighted and binary adjacency matrices.

1.2.2 Degree of a Node

The *degree* of a node in an undirected binary graph is the number of edges connected to it. Mathematically, the degree k_i of node i is the sum of the entries in the ith row of the adjacency matrix:

$$k_i = \sum_{j=1}^{N} A_{ij}. \tag{1.3}$$

In directed graphs, we distinguish between the *in-degree* (the number of edges coming into the node) and the *out-degree* (the number of edges going out from the node) [38].

In weighted graphs, the concept of *strength* generalizes the degree. The strength s_i of a node i is the sum of the weights of the edges connected to it:

$$s_i = \sum_{j=1}^{N} W_{ij}. \tag{1.4}$$

In directed weighted graphs, we similarly distinguish between in-strength and out-strength.

1.2.3 Degree Matrix

We define a *degree matrix* $\mathbf{D}$ as a diagonal matrix, where the D_{ii}th element on the diagonal is the degree of a node i, and all other elements are zeroes. The **degree matrix** plays an important role in subsequently discussing the Laplacian matrix. For a weighted graph, the diagonal elements D_{ii} are then a value of strength for each node i.

For example, consider a simple diagonal matrix $\mathbf{D}$:

$$\mathbf{D} = \begin{pmatrix} d_1 & 0 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 \\ 0 & 0 & d_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & d_n \end{pmatrix}$$

Here, $d_1, d_2, \ldots, d_n$ are the degrees (or strengths in the case of a weighted graph) of the respective nodes.

1.2.4 Laplacian Matrix

Another matrix, also closely related to the adjacency matrix but different in several key ways, also gives some revealing insights into network structure. The *Laplacian matrix* $\mathbf{L}$, commonly referred to as a graph Laplacian, naturally arises when replacing the Laplacian operator, in the diffusion equation, with a matrix form, being in the same form as the ordinary diffusion equation for a gas, but that the Laplacian operator ∇^2 is replaced by the matrix $\mathbf{L}$ [38]. This substitution allows us to model diffusion processes on networks. The matrix $\mathbf{L}$ is called the graph Laplacian for this reason, though its significance extends far beyond diffusion processes. The graph Laplacian also appears in a variety of contexts, including random walks on networks, resistor networks, graph partitioning, and network connectivity [38]. Defining the Laplacian matrix as

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{1.5}$$

where $\mathbf{D}$ is the degree matrix and $\mathbf{A}$ is the adjacency matrix. In weighted networks, the Laplacian matrix is defined using the weighted adjacency matrix:

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \tag{1.6}$$

where $\mathbf{W}$ is the weighted adjacency matrix.

In some contexts, especially when dealing with networks where nodes have varying levels of activity or strength, it is useful to work with the *normalized Laplacian matrix* [11]. This matrix is defined as:

$$\mathcal{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}, \tag{1.7}$$

In this equation $\mathbf{I}$ is the identity matrix and $\mathbf{D}^{-1}\mathbf{A}$ is the normalized adjacency matrix, where the adjacency matrix is scaled by the degree of each node [38].

In weighted graphs, the normalized Laplacian matrix can be defined similarly, using the weighted degree matrix:

$$\mathcal{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}, \tag{1.8}$$

where $\mathbf{W}$ is the weighted adjacency matrix [5].

The Laplacian and normalized Laplacian matrices are important for analyzing network connectivity, diffusion processes, and spectral graph theory [11]. The elements of these matrices reveal how each node relates to the entire network. The normalized Laplacian is particularly useful for studying random walks on networks, where a walker's steps depend on node connectivity. Since random walks are closely linked to diffusion, the normalized Laplacian provides a framework for understanding these dynamics.

1.3 Dynamics and Statistical Physics of Networks

The study of how information spreads in networks is extremely important, with understanding the dynamics of complex networks from the viewpoint of statistical physics. Drawing on concepts drawn from statistical mechanics, one can study how diffusion processes spread through a network, thus indicating in a complex way how structure in a network determines the flow of information. The Laplacian matrix can be very pivotal in that analysis since it allows the modeling of the diffusion process and understanding of how it could differently behave in networks that may appear similar but have subtle structural variations [13]. This chapter will discuss how the architecture of the network affects the diffusion process and extend these concepts to the framework of the density matrix. This topic plays a key role in understanding network dynamics, which can be used for more precise recognition of different networks with different structures.

1.3.1 Diffusion on Networks

One of the key applications of the Laplacian matrix is in modeling *diffusion processes* on networks [18]. Diffusion refers to the process by which something, such as information, heat, or particles, spreads across a network. The general diffusion equation on a network is given by:

$$\frac{\partial \vec{\psi}(\tau)}{\partial \tau} = -\mathbf{L}\vec{\psi}(\tau), \tag{1.9}$$

where $\vec{\psi}(\tau)$ represents a vector in which each element $\psi_i(\tau)$ corresponds to the state (e.g., concentration, temperature) at node i at time τ. The Laplacian matrix $\mathbf{L}$ governs the diffusion process by determining how the state at each node changes over time based on the states of its neighboring nodes [11].

The general solution to this equation is given by:

$$\vec{\psi}(\tau) = e^{-\mathbf{L}\tau}\vec{\psi}(0), \tag{1.10}$$

where $\vec{\psi}(0)$ is the initial state of the system at time $\tau = 0$.

This solution describes how the initial state $\vec{\psi}(0)$ evolves under the influence of the network's structure, as encoded by the Laplacian matrix $\mathbf{L}$. The matrix exponential $e^{-\mathbf{L}\tau}$ effectively acts as the propagator of the diffusion process, determining how the state at each node spreads across the network as time progresses. In weighted networks, this formulation remains valid, with the weighted Laplacian matrix governing the diffusion process while accounting for the varying strengths of connections between nodes [38].

1.3.2 Von Neumann Entropy

The entropy of a random variable quantifies the level of uncertainty associated with its possible outcomes. For a given random variable X, entropy is defined as a function of its probability distribution. A classical measure of entropy is the Shannon entropy [47], given by:

$$H(X) = -\sum_x p_x \log p_x \tag{1.11}$$

where, by convention, $0 \log_2 0 \equiv 0$. This measure intuitively represents the amount of resources needed to encode the information.[41]. Von Neuman entropy is the extension of the classical Shannon entropy which originally developed in the context of quantum systems, quantifies the uncertainty or disorder associated with the state of a system. In quantum mechanics, it is defined for a system represented by a density matrix ρ [21], which is a positive semidefinite, Hermitian matrix with a trace equal to one[13]. The Von Neumann entropy $S(\rho)$ is given by:

$$S(\rho) = -\mathrm{Tr}(\rho \log_2 \rho), \tag{1.12}$$

where $\log_2$ ensures the entropy is measured in bits. With the eigenvalues of the density matrix, the Von Neumann entropy can then be expressed as:

$$S(\rho) = -\sum_i \lambda_i \log_2 \lambda_i, \tag{1.13}$$

which is the Shannon entropy of the eigenvalues of the density matrix, representing the average uncertainty in the system's state.

1.3.3 Density Matrix

The propagator of diffusion processes or random walk dynamics has recently been utilized to define the state of a system, in a manner analogous to the approach widely used in quantum statistical mechanics to define the mixed state of entangled quantum units [13]. In this framework, the state of the network is defined by the density matrix:

$$\boldsymbol{\rho}(\tau) = \frac{e^{-\tau \boldsymbol{\mathcal{L}}}}{Z(\tau)}, \tag{1.14}$$

where $Z(\tau) = \mathrm{Tr}\left(e^{-\tau \boldsymbol{\mathcal{L}}}\right)$ is a normalization factor that plays a role similar to the partition function in statistical mechanics. This state, which is formally identical to a density matrix [21], is then used to calculate the Von Neumann entropy of a complex network.

The corresponding density matrix [13] is defined by:

$$\rho(\tau) = \frac{e^{-\tau \boldsymbol{\mathcal{L}}}}{\mathrm{Tr}\left(e^{-\tau \boldsymbol{\mathcal{L}}}\right)}. \tag{1.15}$$

Note that τ controls the propagation scale. For small values of τ, diffusion occurs primarily between neighboring nodes, whereas larger values of τ facilitate long-range interactions across the network. Furthermore, the density matrix stores important

information about the network at a specific scale. Figure 1.16 represents the importance of communities in difference of density matrices. By using tools like Jensen-Shannon divergence or Trace distance, these matrices can be compared to assess the similarity and mutual information between complex networks, which is a key objective of this thesis.

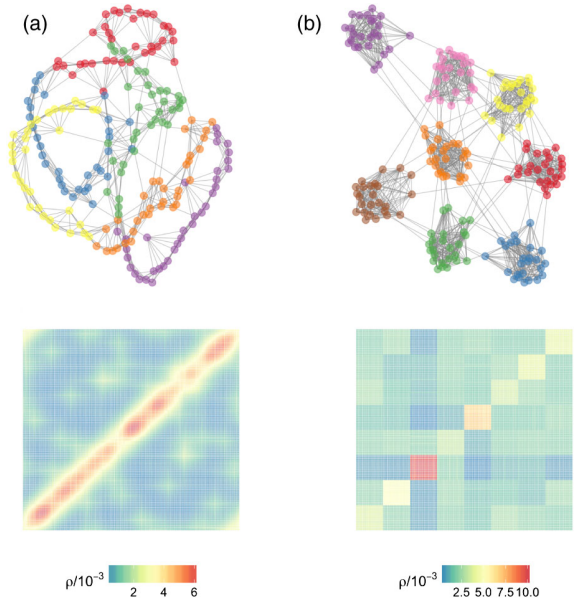

Figure 1.2: Density matrices for two networks. The figure displays the network (top) and the corresponding density matrix (bottom) for (a) a stochastic block model and (b) a Watts-Strogatz network with 200 nodes, at $\tau = 3.16$. In the networks, node colors represent different communities, while in the density matrices, the colors indicate the magnitude of each entry. Figure from Ref. [13].

1.3.4 Spectral Entropy

By using the density matrix of the network in the Von Neumann entropy, we can define spectral entropy[13]:

$$S(\tau) = -\mathrm{Tr}\left(\rho(\tau)\log_2 \rho(\tau)\right). \tag{1.16}$$

Given the eigendecomposition of the Laplacian matrix $L = Q\Lambda Q^{-1}$, where Λ is the diagonal matrix containing the eigenvalues of L, it is straightforward to express the normalization factor Z as:

$$Z = \sum_{i=1}^{N} e^{-\tau\lambda_i(L)}, \tag{1.17}$$

In this equation, $\lambda_i(L)$ represents the ith eigenvalue of L, and the time-like parameter τ cannot take negative values. For $\tau = -1$, this normalization factor yields the Laplacian index of a network, which is relevant in various applications within graph theory and network science [19, 8, 14].

The spectral entropy $S(\rho)$ of the density matrix ρ can be expressed as:

$$S(\rho) = -\sum_{i=1}^{N} \lambda_i(\rho) \log_2 \lambda_i(\rho). \tag{1.18}$$

It follows that:

$$\lambda_i(\rho) = Z^{-1} e^{-\tau \lambda_i(L)}, \tag{1.19}$$

providing the relationship between the eigenvalues of the density matrix and the Laplacian matrix.
This entropy measure quantifies the complexity or disorder within the network by evaluating the uncertainty associated with the distribution of information across the network's structure. A higher spectral entropy indicates a more uniform spread of the network's eigenvalues, corresponding to a more complex or disordered system. Conversely, a lower spectral entropy suggests a more ordered structure, where certain diffusion modes dominate. In other words, the Spectral entropy shows the diversity of information pathways in the system concerning the scale parameter τ.

Spectral entropy provides valuable insights into the structural and dynamic properties of networks, helping to understand how the network topology influences its capacity for information processing and diffusion. For example, networks with homogeneous structures typically exhibit lower spectral entropy, while networks with heterogeneous or more complex structures display higher entropy values. By comparing spectral entropy across various networks, it is possible to gain a deeper understanding of the inherent properties that drive network behavior, making it a powerful tool for network analysis and characterization. In Figure 1.3, the spectral entropies are plotted for three generative models under different configurations. For the Watts-Strogatz model, at certain values of the scale parameter τ (denoted as β in the figure), the models exhibit the same entropy, indicating that the diversity of pathways for the networks is similar at that scale. A significant advantage of spectral entropy is its ability to reveal how entropy varies across different scales of τ, reflecting local and global properties of the network. This approach allows for the observation of differences and similarities between networks at both short and long scales, providing a comprehensive view of network structure and dynamics.

1.4 Generative Models

In network science, generative models are crucial for understanding the mechanisms behind the formation of complex networks. Such models allow one to replicate structural properties that arise in actual networks, such as degree distribution, clustering, or path lengths, leading to better insight into their properties. The basic goal of generative models is to understand the elementary processes that have formed the topology and functional properties of the network. In this section, we will briefly introduce some of the most used models: Erdős-Rényi, Barabási-Albert, Watts-Strogatz, Configuration, and the Exponential Random Graph Model (ERGM).

1.4.1 Erdős–Rényi Model

The Erdős–Rényi (ER) model [17] is one of the most fundamental, and widely studied models in network science, providing a simple way to generate random graphs.

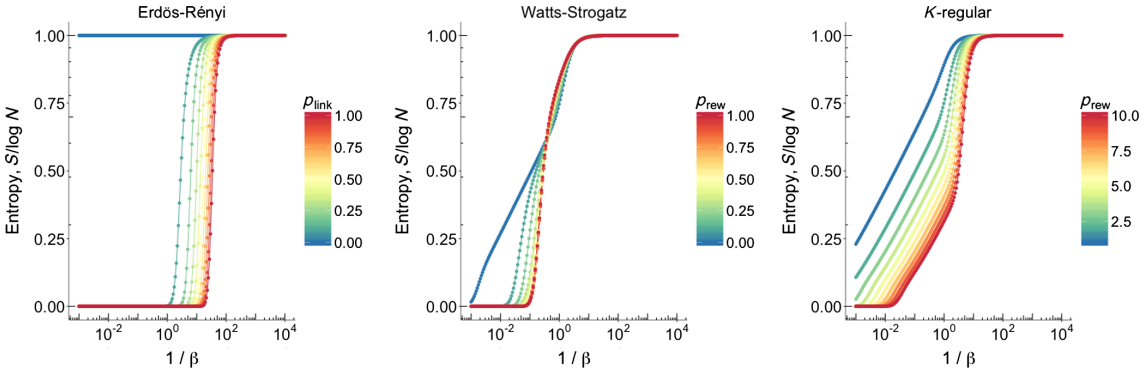

Figure 1.3: Normalized spectral entropies for three types of networks. In this plot, β equals τ. The entropies are shown for Erdös-Rényi networks (left), Watts-Strogatz small-world networks (center), and K-regular lattices (right). The parameters are defined as follows: p_{link} represents the probability of a link between two nodes, p_{rew} is the rewiring probability, and K is the number of neighbors. Figure from Ref.[13]

In the Erdős–Rényi model, a graph $G(N, p)$ is constructed by connecting N nodes, where each possible edge between any pair of nodes exists independently with probability p. Let $G(N, p)$ denote a random graph where N is the total number of nodes. Each pair of nodes (i, j) is connected by an edge with probability p, independently of other pairs.

The degree k of a node in an Erdős–Rényi graph follows a binomial distribution, as each of the $N - 1$ possible edges can be present or absent with probability p:

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k} \tag{1.20}$$

For large N and small p, this distribution can be approximated by a Poisson distribution:

$$P(k) \approx \frac{\lambda^k e^{-\lambda}}{k!} \tag{1.21}$$

where $\lambda = p(N-1)$ is the average degree $\langle k \rangle$ of the network.

The clustering coefficient C, which measures the likelihood that two neighbors of a node are also neighbors, is relatively low in the ER model. For large N, it can be approximated by:

$$C \approx p \tag{1.22}$$

This result indicates that, unlike many real-world networks, the ER model does not exhibit high clustering. The characteristic path length L, representing the average shortest path between any two nodes, scales logarithmically with the number of nodes:

$$L \approx \frac{\log(N)}{\log(\langle k \rangle)} \tag{1.23}$$

This implies that Erdős–Rényi networks can exhibit the small-world [56] property, where most nodes can be reached from any other in a relatively small number of steps.

The Erdős–Rényi model, despite its simplicity, offers a fundamental understanding of random network properties. Its mathematical tractability makes it a valuable tool for studying the emergence of macroscopic properties from microscopic randomness. However, it may not capture the high clustering and scale-free degree distributions typical of many real-world networks, as well as the complex properties that often arise from the presence of hubs, which do not naturally appear in a totally random network.

1.4.2 Barabási-Albert Model

The Barabási-Albert model [5] introduces the concept of preferential attachment, which is key to understanding the emergence of scale-free networks. These networks, ranging from genetic networks to the World Wide Web, exhibit a heterogeneous topology characterized by a power-law distribution of node degrees. The model begins with a small, connected network of m_0 nodes. At each time step, a new node with m edges is added to the network. The probability $\Pi(k_i)$ that the new node connects to an existing node i with degree k_i is given by:

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j}, \tag{1.24}$$

This mechanism leads to a power-law degree distribution:

$$P(k) \sim k^{-\gamma}, \tag{1.25}$$

with an exponent $\gamma = 3$ for the typical case. The Barabási-Albert model effectively explains the emergence of hubs in networks such as the Internet, citation networks, and social networks.

1.4.3 Watts-Strogatz small-world model

Another example of emergence is the small-world property, which is prevalent in various complex systems. These include networks such as brain neurons, electrical power grids, food webs, metabolic pathways, voter connections, telephone communication graphs, and social influence networks. In all of these cases, the average shortest path between any two units is surprisingly short. The Watts-Strogatz model [56] was developed to interpolate between regular lattices and random graphs, capturing the small-world phenomenon. The model starts with a regular ring lattice where each node is connected to k nearest neighbors which can be seen in Figure 1.4. Therefore, with probability p, each edge is randomly rewired, introducing shortcuts that reduce the average path length while preserving a high clustering coefficient.

The average clustering coefficient $C(p)$ and the average shortest path length $L(p)$ as functions of the rewiring probability p are given by:

$$C(p) \approx \frac{3(k-2)}{4(k-1)}(1-p)^3, \tag{1.26}$$

$$L(p) \approx \frac{N}{2k} f(p), \tag{1.27}$$

where $f(p)$ interpolates between 1 and $\log N / \log k$ as p increases. The Watts-Strogatz model successfully reproduces the small-world properties of many real-world networks, characterized by short average path lengths and high clustering.

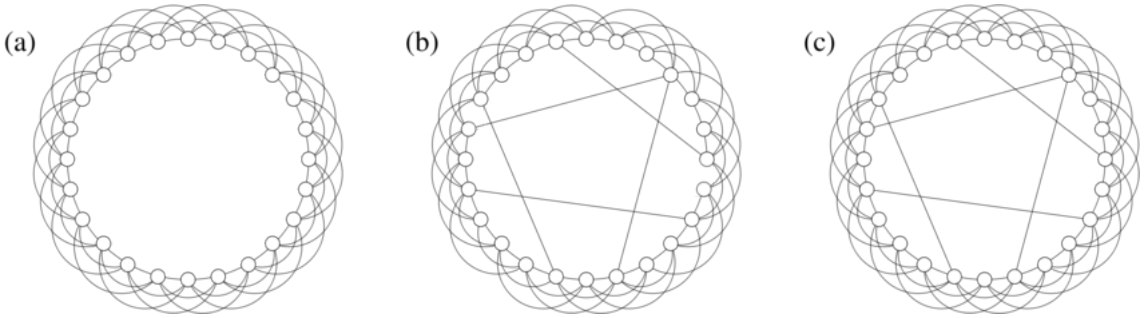

Figure 1.4: Watts-Strogatz Model emerging by rewiring a lattice from Ref [56].

1.4.4 Configuration Model

The Configuration Model allows the construction of random networks by fixing an exact list of degrees, called the degree sequence, to generate the network. Unlike the Erdős–Rényi model, where edges are distributed uniformly at random, the Configuration Model preserves the degree distribution of nodes and in this way, it generates a more realistic model for many real-world networks.

The approach, originally proposed by Molloy and Reed [37], is to consider the desired degree sequence $\{k_1, k_2, \ldots, k_N\}$, which specifies the number of connections each node in the network should have. This degree sequence can be easily extracted from real-world data and used to generate random networks without topological correlations while preserving the degree distribution in a controlled manner. For a given degree sequence, the number of distinct configurations in the ensemble reflects the various ways to connect nodes while maintaining the specified degrees.

The model operates as follows: each node is assigned a number of "stubs" or "half-edges" based on its degree, and these stubs are then randomly paired to form edges. This process ensures that the degree distribution of the resulting network matches the specified degree sequence.

One of the most important properties of the Configuration Model is that it can generate networks without higher-order dependencies, making it typically employed as the null model for studying the impact of the degree distribution on various network properties. The probability that two nodes, i and j, with degrees k_i and k_j, respectively, are connected is given by:

$$P_{ij} = \frac{k_i k_j}{2m}, \tag{1.28}$$

where m is the total number of edges in the network. This simple rule ensures that the expected degree of each node matches the specified degree distribution.

The Configuration Model is particularly useful for generating networks with a broad range of degree distributions, including power-law and exponential distributions, which are often observed in real-world networks. However, its simplicity also leads to some limitations, such as the lack of clustering and the presence of self-loops or multi-edges. Techniques such as edge rewiring can be used to mitigate these issues while preserving the degree distribution.

1.4.5 Exponential Random Graph Models (ERGM)

The other model, which is more flexible, is ERGM. It is possible to generate networks with specific properties by setting those properties as constraints of the model. ERGMs aim to "describe parsimoniously the local selection forces that shape the global structure of a network" [27]. An ERGM would be based on the principle of maximum entropy [28, 29] from statistical physics. This method constructs the most unbiased probability distribution for a given set of constraints, such as the number of edges, degree sequence, or number of triangles in the graph.

Given a network $G(V, E)$ with N nodes, ERGMs consider a set of M network descriptors $\theta_1(G), \theta_2(G), \ldots, \theta_M(G)$. These descriptors can be any measurable properties of the network, such as the number of triangles, the degree sequence, or the presence of specific motifs. If we impose these descriptors as hard constraints, we obtain a microcanonical ensemble where each network in the ensemble strictly satisfies these constraints:

$$\theta_m(G) = \bar{m}, \quad m = 1, 2, \ldots, M \tag{1.29}$$

Conversely, if we relax these constraints and focus on their expected values $\langle \theta_m(G) \rangle$, we generate a canonical ensemble:

$$\sum_{G \in \mathcal{G}} \theta_m(G) P(G) = \langle \theta_m \rangle \equiv \bar{\theta}_m, \quad m = 1, 2, \ldots, M, \tag{1.30}$$

where $\bar{\theta}_m$ represents the target averages of the network descriptors. In this case, networks that closely match the desired values are more likely to be generated than those that deviate significantly.

We need to construct the network ensemble $\mathcal{G}(\mathcal{C})$ such that:

$$\sum_{G \in \mathcal{G}} P(G) = 1, \tag{1.31}$$

We use Lagrange multipliers, which help in optimizing the probability distribution $P(G)$ under the given constraints. The key idea in ERGMs is to find a probability distribution $P(G)$ that maximizes the Shannon entropy:

$$S = -\sum_{G \in \mathcal{G}} P(G) \log P(G), \tag{1.32}$$

This formulation ensures that the ensemble of networks generated by the ERGM reflects the specified topological features while allowing for variability across different network instances.

ERGMs are particularly useful in social network analysis and other fields where networks are expected to exhibit specific structural patterns. By adjusting the parameters associated with the network descriptors, ERGMs can model complex dependencies between nodes, such as homophily (the tendency of similar nodes to connect) and triadic closure (the tendency of two neighbors of a node to be connected).

However, fitting ERGMs to empirical data can be computationally challenging, especially as the number of constraints increases. Despite these challenges, ERGMs remain a valuable tool for understanding and modeling the structure of real-world networks.

1.5 Topological Features of Network

Other than the functional properties that arise from the diffusion processes and the multiplicity of paths, previously discussed, other topological properties of networks are central to understanding the structuration of a complex network. A natural consequence to ask when analyzing a complex network is: which nodes are more important than others? However, relevance cannot clearly be defined, since there is nothing to objectively measure whether someone is relevant or not. Most of the time, it depends on the context and on the way the centrality of a node is subjectively looked at. Moreover, some features describe properties of the whole network, for example, assortativity.

Topological features of networks include node degree, clustering coefficient, betweenness centrality, and closeness centrality. These indexes will be descriptive of the baseline against which complexity and functionality can be described in full detail because they allow getting a detailed perspective on connectivity patterns. Furthermore, they permit comparison among different networks through the study of these key features.

Degree Centrality: Each node in a network has an attribute called *degree*, which is the number of connections (or edges) that it has with other nodes. The degree of a node k_i can be calculated using the adjacency matrix A_{ij}, as shown in Equation 1.3. This equation sums the relevant entries in the adjacency matrix to determine the degree of each node. In the case of weighted networks, where connections have different strengths, we define the concept of *strength*. The strength of a node, analogous to degree in weighted networks, is calculated using a weighted adjacency matrix, as represented in Equation 1.4.

Local Clustering Coefficient: An important topological feature in networks that provides information about the local connectivity of the network is the clustering coefficient. It is a measure that shows the level of cohesion in the neighborhood of a node. This can be divided into local values, which measure the cohesion around a specific node. The local clustering coefficient thus provides insights into the local connectivity structure of the network.

We can define the clustering coefficient for binary networks as:

$$c_i(A) = \frac{1}{2}\sum_{j\neq i}\sum_{h\neq(i,j)} a_{ij}a_{ih}a_{jh}\cdot\frac{1}{\frac{1}{2}k_i(k_i-1)}, \tag{1.33}$$

which simplifies to:

$$c_i(A) = \frac{(A^3)_{ii}}{k_i(k_i-1)} \quad \forall i \in N, \tag{1.34}$$

where A is the adjacency matrix, and $(A^3)_{ii}$ is the ith element of the main diagonal of A^3. The k_i corresponds to the degree of node i [56, 46].

For weighted networks, the standard form of the clustering coefficient considers the strength of the connections:

$$c_i^w = \frac{1}{s_i(k_i-1)}\sum_{j,h}\left(\frac{w_{ij}+w_{ih}}{2}\right)a_{ij}a_{ih}a_{jh}, \tag{1.35}$$

This coefficient is a measure of the local cohesiveness that accounts for the importance of the clustered structure based on the amount of traffic or interaction intensity actually found on the local triplets [6].

Another definition by Onnela et al. [42] proposes a version of the weighted clustering coefficient based on the concept of subgraph intensity, defined as the geometric average of subgraph edge weights. This formulation is used in the NetworkX [24] Python library and has been applied in this thesis, as well as in the work by Suarez et al. [51], resulting in:

$$c_i(W) = \frac{1}{2}\sum_{j\neq i}\sum_{h\neq(i,j)} w_{ij}^{\frac{1}{3}} w_{ih}^{\frac{1}{3}} w_{jh}^{\frac{1}{3}} \cdot \frac{1}{\frac{1}{2}k_i(k_i-1)}, \tag{1.36}$$

simplifies to:

$$c_i(W) = \frac{(W^{\frac{1}{3}})_{ii}^3}{k_i(k_i-1)} \quad \forall i \in N, \tag{1.37}$$

where W is the weighted adjacency matrix.

Betweenness Centrality: Betweenness centrality, originally introduced by Freeman [22], measures the importance of a node based on how frequently it appears on the shortest paths, or geodesics, that connect other nodes in the network. For any pair of nodes j and k, which represent the endpoints of a geodesic path, let σ_{jk} denote the total number of such shortest paths. The term $\sigma_{jk}(i)$ represents the number of these geodesic paths that pass through node i. The betweenness centrality of node i is then defined as:

$$b_i = \frac{1}{(n-1)(n-2)}\sum_{j\neq i}\sum_{h\neq(i,j)} \frac{\sigma_{hj}(i)}{\sigma_{hj}} \quad \forall i \in N, \tag{1.38}$$

where n is the total number of nodes in the network [22, 9, 32].

The definition for betweenness centrality in weighted networks is similar, with the primary difference being that shortest paths are calculated based on the edge weights.

Closeness Centrality: Closeness centrality measures how central a node is within a network by evaluating the mean shortest (geodesic) path length from that node to all other nodes. This metric reflects the node's ability to spread information efficiently across the network.

For binary networks, the closeness centrality $e_i(A)$ is calculated as:

$$e_i(A) = \frac{\sum_{j\neq i}\sum_{h\neq(i,j)} a_{ij}a_{ih}[d_{jh}(N_i)]^{-1}}{k_i(k_i-1)} \quad \forall i \in N, \tag{1.39}$$

where $d_{jh}(N_i)$ is the shortest (geodesic) path length between nodes j and h that includes only the neighbors of i, estimated using the binary adjacency matrix A [34, 46].

For weighted networks, the closeness centrality $e_i(W)$ takes into account the strengths of connections by considering the shortest weighted path:

$$e_i(W) = \frac{\sum_{j\neq i}\sum_{h\neq(i,j)}\left(w_{ij}w_{ih}[d^w_{jh}(N_i)]^{-1}\right)^{\frac{1}{3}}}{k_i(k_i-1)} \quad \forall i \in N, \tag{1.40}$$

where $d^w_{jh}(N_i)$ is the shortest weighted path length between nodes j and h, estimated using the weighted adjacency matrix W [34, 46].

Characteristic Path Length: The characteristic path length L is the average shortest path length between all pairs of nodes in the network [56, 46]. It is defined as:

$$L = \frac{1}{N(N-1)}\sum_{i\neq j} d(i,j), \tag{1.41}$$

where N is the total number of nodes in the network and d_{ij} is the shortest (geodesic) path length between nodes i and j.

For weighted networks, the characteristic path length L^w is given by:

$$L^w = \frac{1}{n}\sum_{i\in N}\sum_{j\neq i}\frac{d^w_{ij}}{n-1}, \tag{1.42}$$

where d^w_{ij} is the shortest (weighted) path length between nodes i and j, and n is the number of nodes in the graph [56, 46].

Transitivity: Transitivity is similar to the clustering coefficient but serves as a global measure in network analysis. It is quantified as the ratio between the observed number of closed triangles and the maximum possible number of such triangles in the network.

For binary networks, transitivity $T(A)$ is defined as:

$$T(A) = \frac{\sum_{i\in N}\sum_{j,h\in N} a_{ij}a_{ih}a_{jh}}{\sum_{i\in N} k_i(k_i-1)} = \frac{\sum_{i\in N}(A^3)_{ii}}{\sum_{i\in N} k_i(k_i-1)}, \tag{1.43}$$

where A is the binary adjacency matrix, and k_i is the degree of node i [40, 46].

We define the standard form of transitivity, which is generalized for multilayer networks, as follows:

$$c(W^\alpha_\beta) = N^{-1}\frac{W^\alpha_\gamma W^\gamma_\delta W^\delta_\beta}{W^\alpha_\gamma F^\gamma_\delta W^\delta_\beta}, \tag{1.44}$$

where W^α_β represents the weighted adjacency matrix, and F^γ_δ corresponds to a complete network without self-loops. The term N^{-1} is a normalization factor that ensures the clustering coefficient is scaled appropriately [15].

Additionally, there is a definition of non-standard transitivity $T(W)$ used in the work of Suarez et al. [51], which measures the 'intensity' of observed closed triangles relative to the maximum possible 'intensity' of closed triangles. This non-standard transitivity is defined as:

$$T(W) = \frac{\sum_{i\in N}\sum_{j,h\in N} w_{ij}^{\frac{1}{3}}w_{ih}^{\frac{1}{3}}w_{jh}^{\frac{1}{3}}}{\sum_{i\in N} k_i(k_i-1)} = \frac{\sum_{i\in N}(W^{\frac{1}{3}})^3_{ii}}{\sum_{i\in N} k_i(k_i-1)}, \tag{1.45}$$

where W is the weighted adjacency matrix.

Assortativity: Assortativity measures the correlation between the degree of a node and the mean degree of its neighbors. It reflects the tendency of nodes to connect with other nodes that have a similar degree.

For binary networks, assortativity r is defined as:

$$r = \frac{l^{-1}\sum_{(i,j)\in L} k_i k_j - \left[l^{-1}\sum_{(i,j)\in L} \frac{1}{2}(k_i + k_j)\right]^2}{l^{-1}\sum_{(i,j)\in L} \frac{1}{2}(k_i^2 + k_j^2) - \left[l^{-1}\sum_{(i,j)\in L} \frac{1}{2}(k_i + k_j)\right]^2}, \tag{1.46}$$

where l^{-1} is the inverse of the number of links in the graph, k_i is the degree of node i, and L is the set of links in the graph [39, 46].

For weighted networks, assortativity r^w is defined as:

$$r^w = \frac{l^{-1}\sum_{(i,j)\in L} w_{ij} s_i s_j - \left[l^{-1}\sum_{(i,j)\in L} \frac{1}{2} w_{ij}(s_i + s_j)\right]^2}{l^{-1}\sum_{(i,j)\in L} \frac{1}{2} w_{ij}(s_i^2 + s_j^2) - \left[l^{-1}\sum_{(i,j)\in L} \frac{1}{2} w_{ij}(s_i + s_j)\right]^2}, \tag{1.47}$$

where s_i is the weighted degree of node i, and the other variables are defined as above [35, 46].

1.5.1 Summary

We began the chapter by exploring complex systems and discussing how modeling such systems as networks, using graph theory, provides insights into their structure and function. We also examined how statistical mechanics, particularly through the use of the Laplacian matrix, can model diffusion processes within networks. Specifically, we introduced the density matrix as an effective tool for studying information flow in networks. By analyzing spectral entropy at different levels of detail, we demonstrated how the variety of connections within a network influences its overall behavior and functionality. This approach helps uncover the complex properties of the network by considering how its structure varies across different scales.

Next, we reviewed various generative models—such as the Erdős–Rényi model, the Barabási-Albert model, the Watts-Strogatz small-world model, the Configuration model, and Exponential Random Graph Models—applied to represent and analyze real-world networks. These models enhance our understanding of the behaviors and structures observed in empirical networks.

We then discussed the topological properties of networks, including node degree, clustering coefficient, betweenness centrality, closeness centrality, transitivity, and assortativity. These topological properties provide methods to describe networks and compare multiple networks, aiding in the identification of patterns and mutual information.

These tools help us better understand real-world networks and analyze complex systems. They also allow us to determine which models best fit the empirical networks observed in nature. In the next chapter, we will explore the brain as a complex system and network. We will apply these tools to actual connectome data. This will enhance our understanding of brain connectivity and enable comparisons between different connectomes. We will also work on a method to identify the generative models that best represent these real networks.

Chapter 2

Brain Networks and Applications

One of the most fascinating questions is how the brain functions. How do we observe, behave, experience consciousness, and recall memories? All of these capabilities arise from the brain. To answer this question, we must also understand how the brain develops and how evolution has shaped such a complex system. How do DNA codes contribute to the brain's structure across different species? The brain is the most complex organ in most living systems, and its high complexity makes it difficult to analyze thoroughly. One successful approach to studying the brain is to model it as a network and analyze it as a physical system. In this chapter, we will explore the brain and the characteristics that make it a complex system. We will then delve into modeling the brain as a network and conclude by discussing an important dataset of mammalian connectomes.

2.1 Introduction to Brain Networks

Brain networks consist of neurons and synapses that form a complex network capable of executing a wide range of tasks. Despite constituting a small percentage of the total body mass, the brain is the central organ responsible for managing vital functions and coordinating the activities of all other organs to sustain life. The ability to compact a vast array of functions into such a small region necessitates a high level of complexity, enabling the brain to efficiently store, process, and analyze large amounts of information while maintaining robustness and adaptability.

2.1.1 The Brain as a Complex Network

From recent studies, we know the adult human brain contains around 86 billion neurons, with a nearly equal number of non-neuronal cells. Only about 20-25% of these neurons are located in the cortical gray matter, which constitutes roughly 80% of the brain's mass. Around 69 billion which is about 80% of the brain's total neurons, are found in the cerebellum [4, 26]. These numbers vary among different species, but they provide a general sense of the elements within this system.

However, another point besides the huge number of neurons that makes this complex network efficient is how they interact. The brain's network architecture has several key properties of complex networks:

- **Small-World**: The brain has a small-world design, which means it features high local clustering—neurons tend to form tight-knit groups—combined with

short path lengths between any two neurons. This setup promotes efficient communication throughout the entire network [10].

- **Modularity**: The brain is organized into modules or communities, where groups of neurons or brain regions are densely interconnected but have fewer connections to other groups. This modular organization allows for specialized functions in different brain areas while still supporting integration and communication between these modules [36].

- **Scale-Free**: The brain networks having scale-free properties. That means a small number of neurons, are hub and they have a very high number of connections, while most neurons have relatively few. These hubs are crucial for maintaining network stability and facilitating communication across different brain regions [50].

- **Robustness**: The brain is robust and adaptable. It can maintain function despite damage or disruption and can reconfigure itself in response to learning, experience, and injury. This plasticity is a defining feature of complex networks, allowing the brain to adapt to changing environments and demands [49].

This compact yet powerful system requires a highly organized structure to manage vast amounts of information. The brain's network architecture enables efficient data processing and storage, making it both resilient and adaptable to life's challenges. Throughout the animal kingdom, having a brain or central nervous system is a key sign of advanced life, showing just how important neural networks are in shaping complex behaviors and functions.

2.1.2 Biological Basis and Mechanisms of Brain Connectivity

The brain's communication system is a complex cell network that uses electrical and chemical signals to transfer messages using different neurotransmitters, both excitatory and inhibitory. Inotropic neurotransmitters are those that enhance the likelihood of a neuron firing by making the membrane potential more positive, thus increasing the probability of an action potential. On the other hand, depressant neurotransmitters like gamma-aminobutyric acid (GABA) decrease the membrane potential and thereby make it hard for an action potential to be created. The ion gradient which is controlled by ion channels and pumps is important to ensure that neurons interpret the signals they receive from among the many among the inputs [30].

Synaptic connection is the basis of the brain's connectivity and is the key to its information processing, behavioral regulation, and cognitive functions including memory, perception, and decision-making. These connections enable the neurons to interact in order to enable the brain to handle large amounts of information.

In the past few years, neuroscience has started to represent the brain in terms of a network, where nodes are neurons or brain areas and edges are the connections between them. This network-based view unveils the inherent characteristics of the structural and functional brain organization such as small-worldness, modularity, and scale-freeness that allow the brain to function effectively and adaptively [10, 36, 50, 49].

But as Papo and Buldú mentioned in their paper [43], one must question whether these network models are actually capturing the brain's functional architecture or are used for methodological convenience. Apart from identifying connections, it is necessary to look into how these connections support the properties that make the brain a complex network system.

	Basic network terminology in neuroscience
Brain network	Any representation of the brain anatomy or activity with a predefined (global or partial) parcellation, where each brain region corresponds to a node and connections between nodes are quantified according to a certain anatomical or dynamical criterion.
Brain node	A node i represents any brain partition that may have an anatomical or functional meaning, from a single neuron or a voxel, to large brain regions or regions of interest (ROI).
Brain link	A link $\{i, j\}$ quantifies any kind of interaction or relation between two (or more) brain nodes i and j. These can be physical (from synapses to large tracts) or dynamical (e.g., coordinated dynamics between two nodes), evolve with time, and co-exist at different temporal and spatial scales.
Brain connectivity	Brain connectivity is generally defined in the anatomical space and can refer to the presence of anatomical fibers, or statistical dependencies between the activity of different nodes or causal interactions ("effective connectivity") between units identified with nodes within the nervous system.
Sparsity	A network with N vertices and K edges is said to be sparse if K is much smaller than the possible maximum number of links a network with N nodes in principle can have, and the node degree is finite for $N \to \infty$.

Table 2.1: Basic network terminology: constituents and connectivity. Table from Ref [43].

2.2 Network Modeling in the Brain

The brain's complex functionality arises from an intricate web of connections between neurons and brain regions. To comprehend this complexity, advanced modeling methods are essential. Neuroscience has embraced network modeling, which allows scientists to represent the brain's structure and function in a systematic and quantifiable way. This approach provides valuable insights into how different brain regions interact and contribute to overall brain function. However, the accuracy and effectiveness of network modeling heavily depend on the quality of data captured.

2.2.1 Network Representation of Brain Data

When considering brain data obtained using techniques such as functional Magnetic Resonance Imaging (fMRI) [23]and Diffusion Tensor Imaging (DTI)[7], representation as networks might be seen to work effectively. In these networks, nodes represent different parts of the brain while their edges indicate connectivity between regions. Such connections can either be determined by structural data that keep track of physical relationships among regions, or functional data capturing neural activity

patterns over various regions with time. The approach of network representation permits analysis of the brain's wiring and its impact on cognitive processes.

2.2.2 Structural and Functional Brain Networks

Structural brain networks are based on the actual links between neurons or even cortical areas, typically derived from Diffusion Tensor Imaging (DTI). DTI is an advanced imaging modality that uses the Brownian motion of water molecules to provide data for images, representing how neural signals travel through anatomical pathways. On the other hand, functional brain networks, created from Functional Magnetic Resonance Imaging (fMRI), are built upon the temporal correlations of activity observed across various parts of the brain. While structural networks reflect how the brain is wired, functional networks, with their time-varying properties during cognitive processing, inform us about how different components within the brain interact with each other.

2.2.3 Common Metrics and Tools for Brain Network Analysis

Recently, metrics from network science have been commonly used in neuroscience. These include modularity, which measures the degree to which a network can be divided into distinct modules; clustering coefficient, which quantifies the tendency of nodes to form tightly-knit groups; and network efficiency, which assesses the ease of information transfer across the network. Suarez et al. [51] utilized topological features obtained from the mammalian connectomes to compare them. Similarly, Vazza et al. [53] used classical metrics like clustering and modularity to demonstrate the similarity between brain networks and cosmic networks. Tools such as graph theory software and specialized neuroimaging analysis packages allow researchers to apply these metrics to brain data, facilitating the exploration of network properties and their relationship to cognitive functions.

2.3 Mammalian Connectomes Data

Mammalian taxonomies have traditionally been defined by morphological traits and genetics, but how these species differ in terms of neural circuits remains largely unexplored. The primary challenge has been the inconsistency in network reconstruction techniques across species, leading to non-comparable connectomes. To address this, we utilize the Mammalian MRI (MaMI) dataset, which applies a unified MRI protocol to chart connectome organization across 124 species, enabling a systematic comparison of neural architectures within the mammalian phylogenetic spectrum.

2.3.1 Phylogenetic Tree

A phylogenetic tree is a map of the tree of evolutionary relationships of various species on the basis of the existence of certain genetic and physical features among these species. These trees help to explain how species have evolved from a common stock to differentiate from each other in the course of evolution. The branching points on the tree are known as nodes, which are equivalent to the nodes on a family tree, and refer to common ancestors in a species. Trees are created by employing

morphological traits and genetic facts and are indispensable to paleontologists representing the relationships of distinct styles among species. [16]

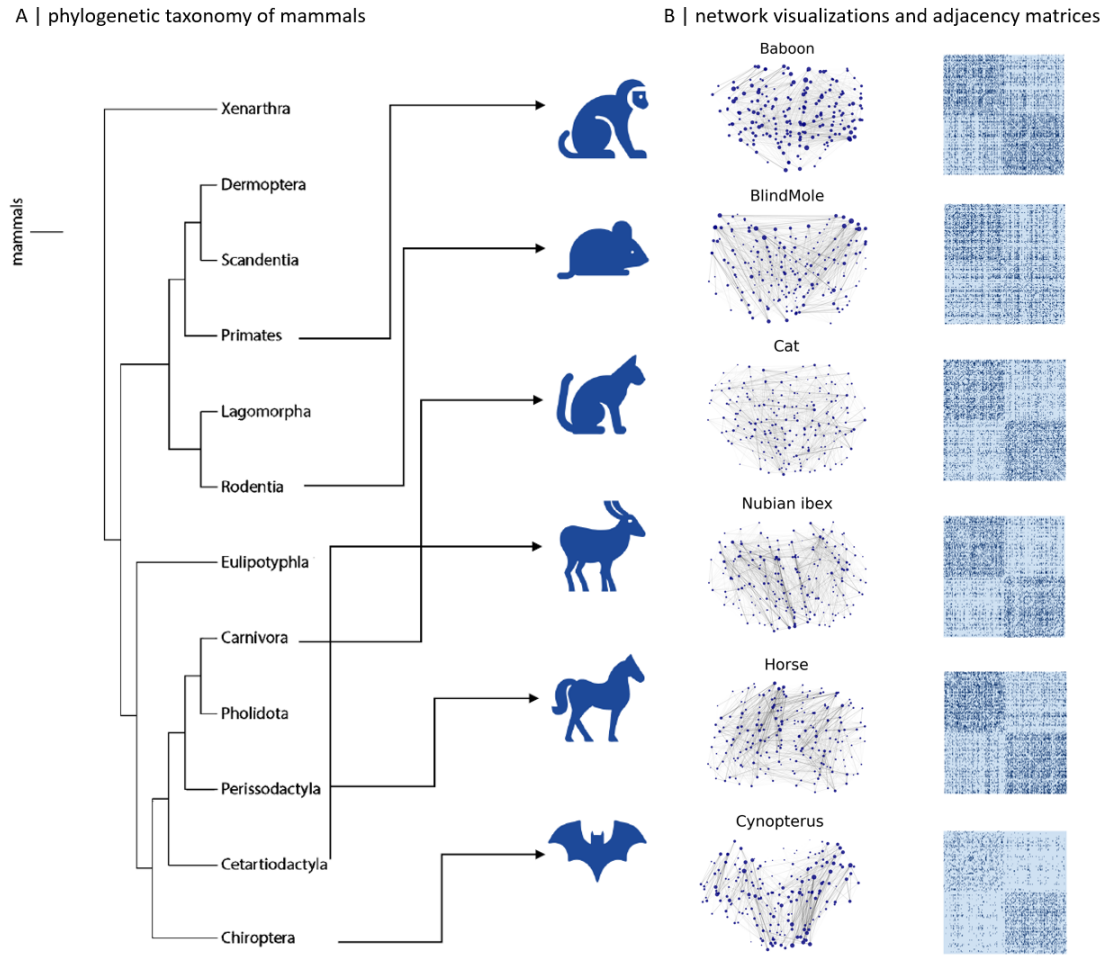

Figure 2.1: Mammalian MRI (MaMI) dataset. (A) illustrates the taxonomy derived from the phylogenetic tree. Only orders containing at least five different connectomes in the dataset were included in the analyses. (B) displays the network visualizations with respect to the node positions in the connectome, alongside the binarized adjacency matrices for the selected species. The left and right cerebral hemispheres are represented in both the network visualizations and the adjacency matrices.

2.3.2 MAMI Dataset

The Mammalian MRI (MAMI) dataset [52] encompasses high-resolution ex vivo structural and diffusion MRI scans of 124 animal species(a total of 225 scans including replicas) spanning 12 morphologically and phylogenetically defined taxonomic orders: Cetartiodactyla, Carnivora, Chiroptera, Eulipotyphla, Hyracoidea, Lagomorpha, Marsupialia, Perissodactyla, Primates, Rodentia, Scandentia, and Xenarthra. Only orders with at least five different connectomes were included in the analyses, represented in Figure 2.1. This dataset is particularly valuable for studying the connectivity patterns in mammalian brains, as it offers a rich source of data for constructing and analyzing brain networks. The uniform imaging protocol and detailed structural data make the MAMI dataset an essential resource for researchers

investigating the principles of brain organization across different species [51]. The dataset's structural MRI scans are parcellated into 200 nodes using k-mean clustering, allowing for the creation of sparse and weighted adjacency matrices that represent brain connectivity. Additionally, the spatial information for these nodes is available, indicating which part of the brain each node corresponds to.

2.3.3 Challenges and Considerations in Using MAMI Data

Using the MAMI dataset for network analysis presents several challenges. These include potential noise and artifacts in imaging data, variations in resolution across species, and biases introduced during data collection. The most important problem is decreasing the resolution from the hierarchy containing about 80 billion neurons down to 200 nodes means the absence of many details and information. The uniform parcellation scheme, which makes it possible to make comparisons, may mask the differences at the level of regions that may be important in a biological sense.

2.4 Summary

In the previous chapter, we introduced network modeling tools that are essential for analyzing complex systems. We then explored brain networks and how connectomes can be modeled as complex networks. This chapter focused on the Mammalian MRI (MaMI) dataset. The dataset includes connectomes from various mammalian species, derived from genetic and phylogenetic data. We discussed the challenges and considerations involved in using this dataset for network analysis.

In the next chapter, we will apply the network modeling tools to the MaMI dataset. We will model these networks to analyze and cluster them based on their functional and topological features. Our goal is to compare the resulting clusters with the phylogenetic tree taxonomy. This will provide insights into the relationship between neural architecture and evolutionary lineage.

Chapter 3

Network Comparison of Mammalian Connectomes Using Functional and Topological Features

Network models of anatomical connections help extract quantitative features that describe brain organization. These features allow for connectome comparisons across different species and orders. Such comparisons can enhance our understanding of brain architecture differences between species and link these differences to existing taxonomies and phylogenies. In this chapter, we use topological and functional tools from network science and statistical mechanics to compute distance matrices of mammalian connectomes. We then compare these matrices to the phylogenetic taxonomy to determine if species within the same order have similar brain networks. Additionally, we apply clustering algorithms to create a taxonomy based on the connectomes and compare it to the phylogenetic tree. To validate the functional method introduced, we first test it on networks generated by random models with similar properties.

3.1 Comparison of Different Groups of Test Networks from Generative Models

The main objective of this thesis is to develop an accurate pipeline for network comparison. We begin by testing some networks through two experiments using generative models. Specifically, we employ the Erdős–Rényi (ER) and Barabási–Albert (BA) models to create networks with varying properties. In Sections 1.4.1 and 1.4.2, we explained how these generative models create random networks.

In the first experiment, we generate five groups of networks using the ER model by adjusting the parameter p. Each group contains five networks with the same parameter value. We then apply a method from statistical mechanics, known as the density matrix, to calculate the distance matrix, which quantifies the differences between networks with distinct properties. We perform a similar experiment with the BA model, adjusting the preferential attachment parameter m to generate and compare the networks.

3.1.1 Selection of Test Networks from Generative Models

To explore the effectiveness of the method in distinguishing various networks, we selected five domains of connectivity for the Erdős–Rényi model around its critical point of connectivity, which is $p = \frac{1}{n}$. In our case, the networks have 100 nodes, so the critical p is 0.01. We also considered $p = \frac{\log n}{n}$, which almost surely results in the formation of a giant connected component (GCC). At this value of p, which is 0.046 in our case, the probability of not having a GCC decreases exponentially. A giant connected component (GCC) is a large subgraph within a network where all nodes are interconnected, typically forming during a percolation process when the connection probability exceeds a critical threshold, allowing a significant portion of the graph to become fully connected. For the Barabási–Albert model, we selected values for m (the preferential attachment parameter) ranging from 1 to 5. The values should be an integer because of the meaning behind the growth rule of the Barabási–Albert model. These parameters p and m are both controlling the connectivity of the network.

3.1.2 Calculation of Density Matrices for Functional Analysis of Test Networks

For each test network, we calculated density matrices at multiple scales. These matrices represent the flow of information across different scales, which is crucial for capturing the functional characteristics of networks. As explained earlier, to obtain the density matrix from Equation 1.14, we first need to calculate the Laplacian matrix. The Laplacian matrix is defined as the degree matrix minus the adjacency matrix (see Equation 1.5). The scale parameter τ is particularly important because it reveals the properties of networks locally to globally, highlighting the powerful capabilities provided by this tool from statistical mechanics.

The use of density matrices allows for a nuanced comparison of networks beyond traditional topological metrics. By applying this method to the selected test networks, we can observe the method's sensitivity to connectivity within networks.

3.1.3 Jensen-Shannon Divergence (JSD) for Calculating Distance Matrix

To quantify the similarity between networks, we employed the Jensen-Shannon Divergence (JSD) between density matrices[13]. The JSD provides a symmetric measure of divergence that is particularly suited for comparing probability distributions.

The Kullback-Leibler entropy divergence between two density matrices $\hat{\rho}(\tau)$ and $\hat{\rho}'(\tau)$ is given by:

$$D_{KL}(\hat{\rho}(\tau)\|\hat{\rho}'(\tau)) = Tr[\hat{\rho}(\tau)(\log\hat{\rho}(\tau) - \log\hat{\rho}'(\tau))],$$

and the Jensen-Shannon divergence is given by:

$$\mathcal{J}(\hat{\rho}(\tau)\|\hat{\rho}'(\tau)) = \frac{D_{KL}(\hat{\rho}(\tau)\|\hat{\rho}''(\tau)) + D_{KL}(\hat{\rho}'(\tau)\|\hat{\rho}''(\tau))}{2},$$

where

$$\hat{\rho}''(\tau) = \frac{\hat{\rho}(\tau) + \hat{\rho}'(\tau)}{2},$$

whose square root provides a symmetric metric for quantifying network dissimilarity.

The resulting distance matrices were used to assess the differences between networks at various scales. We obtained the distance matrices for various values of τ by calculating the pairwise distances between each pair of networks. The distance matrices for the Barabási–Albert model are presented in Figure 3.3 part B, and those for the Erdős–Rényi model are shown in Figure 3.2part B. As can be seen in the figures, networks with different configurations are more distant from each other, while networks generated with the same parameters are closer together. These results can be seen especially in $\tau = 1$.

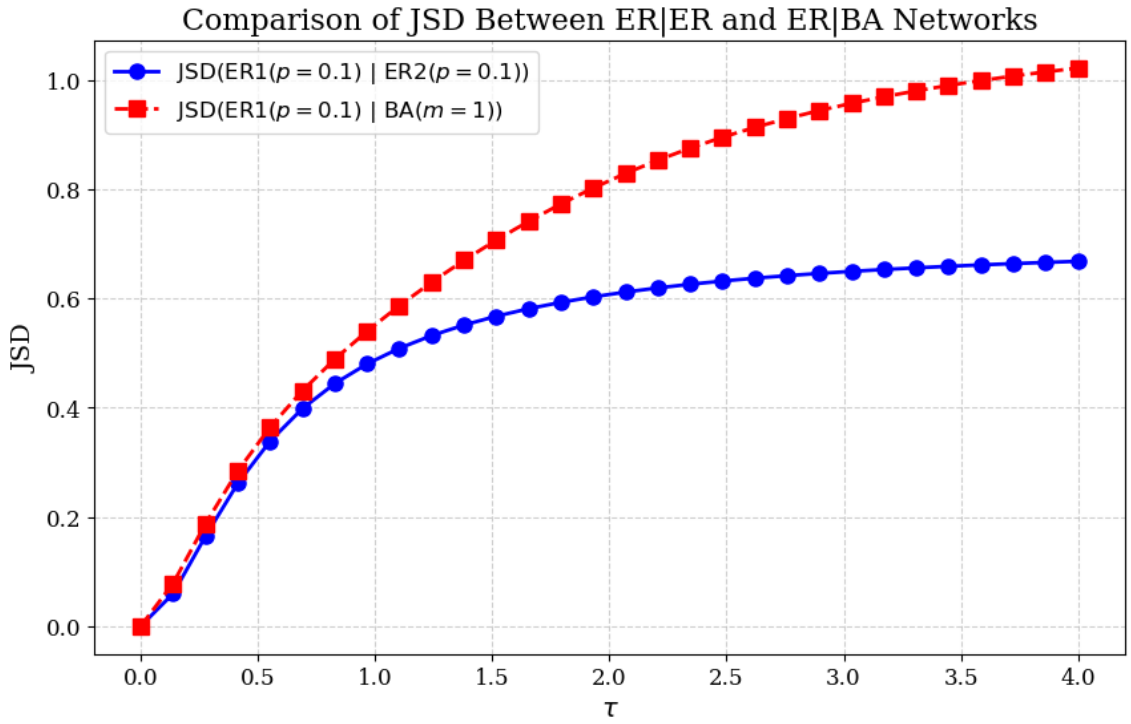

Figure 3.1: Comparison of JSD between ER and BA networks at different scales. The blue line represents the JSD of density matrices between two different ER networks, each with 100 nodes and the same connection probability $p = 0.1$, which is the critical value. The red line shows the JSD between one of the ER networks and a BA network with a preferential attachment parameter equal to 1, across different scales. The networks from the same growth model and configuration are more similar, especially at the long-range of scale parameter τ.

We performed another experiment to demonstrate the effectiveness of the density matrix and JSD for the comparison. We generated three networks: two ER networks with $p = 0.1$ and one BA network with $m = 1$, each with 100 nodes. We then calculated the density matrices at different scales and compared the JSD between the density matrices of the ER networks (JSD(ER1|ER2)) and the JSD between one of the ER networks and the BA network (JSD(ER1|BA)). It can be observed in Figure 3.1 that the blue curve, representing the JSD between the two ER networks, is

consistently lower than the red curve, representing the JSD between the ER and BA networks, across all scales. Additionally, the difference between the curves increases as the scale increases.

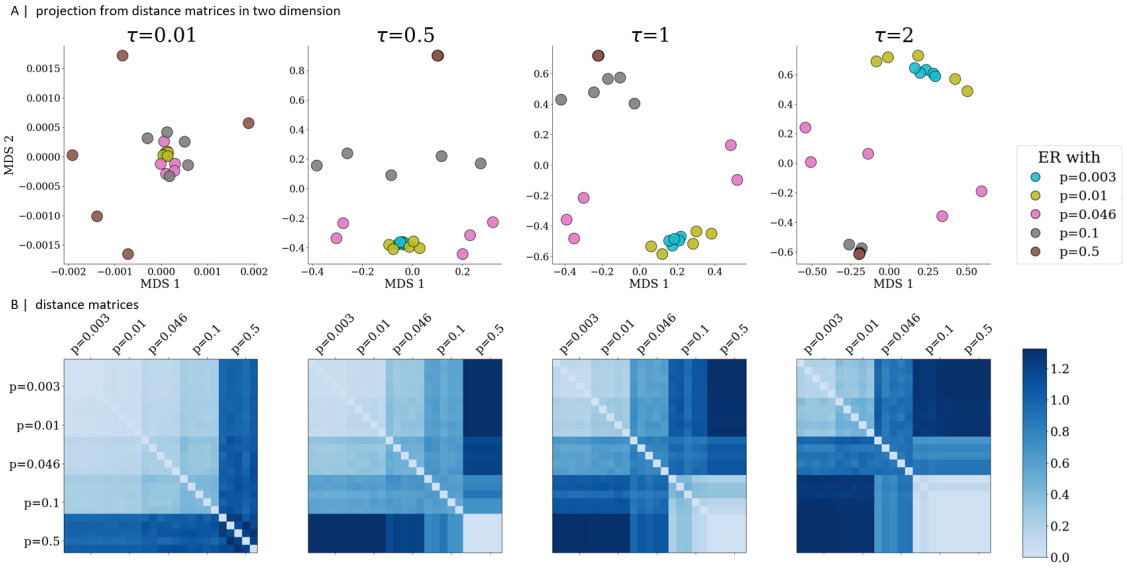

Figure 3.2: Functional distance matrices at different scales of the Erdős–Rényi networks and the projection in two-dimensional space. The comparison involves 25 ER networks, divided into 5 groups with different connectivity probabilities, across four different scales. Part B shows the distance matrices obtained from the density matrices at four scales: $\tau = 0.01$, 0.5, 1, and 2. Part A displays the projection of these distance matrices into a two-dimensional Euclidean space for each scale. The colors represent the networks within each configuration group.

3.1.4 Projecting Distance Matrices of Test Models Using Multidimensional Scaling (MDS)

After obtaining the distance matrices by calculating the JSD for each pairwise comparison of networks, we can utilize Multidimensional Scaling (MDS) to project the distance matrices into an Euclidean n-dimensional space. MDS preserves the pairwise distances between data points as much as possible, allowing for meaningful visualization of complex data structures. By taking a distance or dissimilarity matrix as input, MDS positions data points in a way that the distances in the lower-dimensional space approximate the original distances.

We projected the distance matrices obtained from test networks into a two-dimensional Euclidean space for better visualization using the MDS implementation from the scikit-learn Python library [44]. Figure 3.2 part A shows the projection of the distance matrices of ER networks at different scales. It can be seen that as the τ value of the density matrices increases, the difference between high-connectivity and low-connectivity networks becomes more significant. At $\tau = 2$, the networks are divided into three regimes: one is sparse networks with low connectivity below the critical point, the second is the $p = 0.046$ regime, which is the percolation point where the giant connected component (GCC) begins to form and small islands

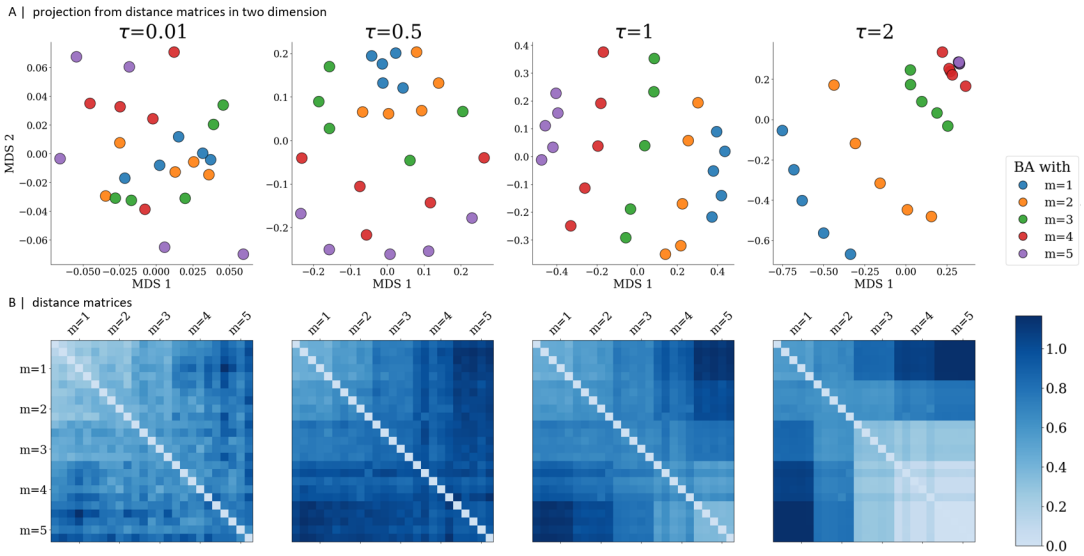

Figure 3.3: Functional distance matrices at different scales of the Barabási–Albert network and the projection into two dimensions. The comparison involves 25 BA networks, divided into 5 groups with different preferential attachment parameters, across four different scales. Part B shows the distance matrices obtained from the density matrices at four scales: $\tau = 0.01, 0.5, 1$, and 2. Part A displays the projection of these distance matrices into a two-dimensional Euclidean space for each scale. The colors represent the networks within each configuration group.

appear in the network, and the last regime consists of the more connected networks with higher p values.

Figure 3.3 part A shows the projection of the distance matrices of BA networks at different scales. In this experiment, it can also be observed that as the τ value of the density matrices increases, the difference between high-connectivity and low-connectivity networks becomes more pronounced. Starting from $\tau = 1$, the networks are divided into five groups due to having the same configuration of the growth model from which they were generated. Networks with the same preferential attachment parameter are closer to each other, and by $\tau = 2$, these groups become clearly distinguishable.

From this experiment, we can understand that the properties of the network, especially connectivity, play a crucial role in the information encoded into the density matrices. Furthermore, by finding an optimal scale parameter that aligns with the comparison we aim to make, we can analyze the networks based on the specific properties we are interested in from a functional perspective. This approach allows us to explore network characteristics beyond traditional topological measures, as the functional method enables us to adjust the scale parameter to extract the desired information.

3.2 Application to the Mammalian Connectomes

In this section, we will apply the methods and pipelines to develop a taxonomy, based on connectomes. In the paper by Suarez et al. [51], it was demonstrated that

certain topological features reveal both intra-order similarities. Our objective is to compare the clustering and distance matrices obtained from functional analysis with the phylogenetic taxonomy to assess how much they align with each other. Additionally, we will compare these results with those obtained from topological analysis.

We will replicate the topological features used by Suarez et al. [51] and introduce two methods for functional analysis: one using density matrices and trace distance, and the other using entropy and free energy profiles. Finally, we will apply clustering and calculate the NMI and AMI scores to assess how closely the clustering matches the phylogenetic tree, and compare the results across the methods.

3.2.1 Overview of Networks in the MAMI Dataset

As discussed in Section 2.3.2, we used the MAMI dataset, which contains 225 mammalian connectomes. For this study, we selected 203 networks from this dataset, focusing on six orders of species: Cetartiodactyla (40), Rodentia (29), Chiroptera (30), Perissodactyla (7), Carnivora (51), and Primates (46). These orders were chosen because the dataset includes more than five networks for each of them (see Figure 2.1). The 203 networks we analyzed are weighted, with each network consisting of 200 nodes. Additionally, we binarized the data by applying a threshold of 1, meaning that if there is a connection in the weighted network, the nodes are considered connected; otherwise, they are not. To identify the most similar connectome-based clustering we explore different methods and use both weighted and binarized networks.

3.2.2 Calculating the Functional Distance Matrix Using Trace Distance

Using Jensen-Shannon Divergence (JSD) for network comparison and obtaining distance matrix is computationally intensive, particularly when dealing with 203 networks. Moreover, JSD does not effectively capture the order of similarities in the weighted networks of the MAMI dataset. We have borrowed the trace distance from quantum mechanics as an alternative metric for calculating distance matrices[41]. The trace distance provides a different perspective on network similarity in our dataset and offers a more robust basis for clustering. The trace distance between two quantum states, represented by density matrices ρ and ρ', is defined as:

$$T(\rho, \rho') := \frac{1}{2}\|\rho - \rho'\|_1 = \frac{1}{2}\mathrm{Tr}\left[\sqrt{(\rho - \rho')^\dagger(\rho - \rho')}\right]$$

This metric measures the distinguishability between the quantum states ρ and ρ', with values ranging from 0 for identical states to 1 for completely distinguishable states. The pipeline for obtaining the distance matrix from the connectomes using trace distance is represented in Figure 3.4. In part A of this figure, we obtain density matrices at scale τ. In Part B, we calculate the trace distance for each pair of connectomes to construct the distance matrix. This matrix is symmetric, and the diagonal elements are zero because the trace distance of each density matrix to itself is zero, indicating that they are identical.

The scale at which we calculate the density matrices is crucial, as each scale τ reflects different properties of the connectomes. Identifying an optimal τ that best

aligns with the taxonomy of species in the phylogenetic tree is essential. To achieve this, we calculated the distance matrix for each value of τ and assessed the Normalized Mutual Information (NMI) and Adjusted Mutual Information (AMI) scores to quantify clustering performance relative to the phylogenetic tree. Calculating the distance matrix for 203 networks requires 20,706 computations. Even with parallelization using the `joblib` library in Python, the computation is still intensive, particularly when exploring various values of τ. The exploration of different values for τ was challenging due to the computational demands. . Therefore, we conducted the exploration on a smaller subset of data. By determining the optimal boundary for τ, we examined various scale parameters with the entire dataset through fewer experiments. Ultimately, we found that a value of $\tau = 0.025$ was the most effective for this clustering.

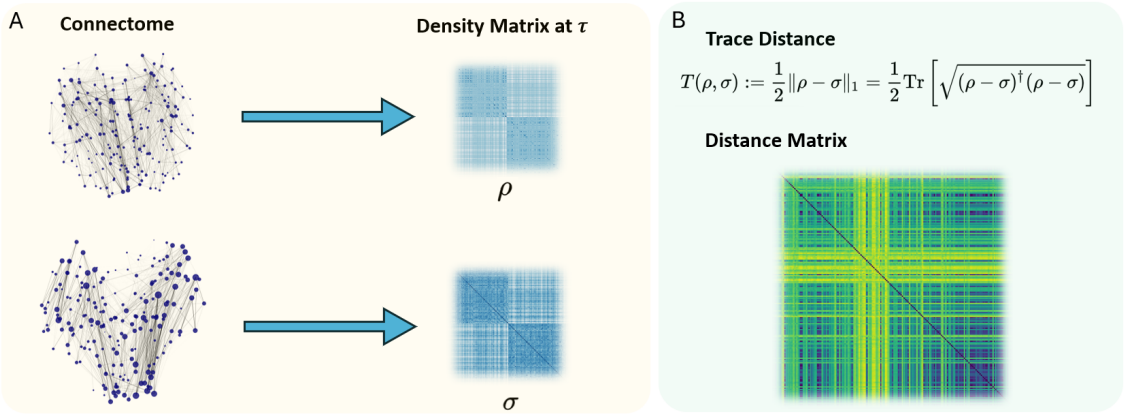

Figure 3.4: Pipeline for obtaining distance matrices using the trace distance. In part A, density matrices are derived from the connectomes at scale τ. In part B, the distance matrix is computed by comparing each pair of density matrices.

3.2.3 Spectral Entropy and Free Energy Curves as Functional Features

Another aspect of the network analysis involves studying their spectral entropy curves. In this section, we examine information pathways at different scales and plot the corresponding entropy curves. From spectral entropy analyses, we can extract an important feature called the **entropy profile** and the **free energy profile**. These features are vectors of entropies and free energies at different scales τ. The advantage of this method is that it is faster because we do not need to work with density matrices; instead, we only need the eigenvalues of the Laplacian matrix (see Equation 1.19) to obtain the entropy profile and free energy profile. Another significant advantage is that we capture all the information across different scales of a network in a single vector. This allows for clustering and comparison by considering short- to long-range τ scales. After obtaining the feature vectors for each network, we can calculate the distance matrix using the cosine dissimilarity. Cosine dissimilarity measures the difference between two vectors by calculating the cosine of the angle between them, with values closer to 1 indicating greater dissimilarity and values closer to 0 indicating greater similarity. Additionally, we can project the vectors into n-dimensional space using MDS or UMAP. These vectors can be

entropy or free energy on different scales. We also tried the feature vector with both. Figure 3.5 represents the mean spectral entropies and free energy corresponding to the phylogenetic tree for both weighted and binarized networks, based on their taxonomy. Each connectome species has an individual curve, and by averaging across each scale parameter τ, we obtain six curves corresponding to the six orders of species. The figure also displays the standard deviation at each scale. We can see the difference of curves showing the difference of information pathways in different scales. Chiroptera, which exhibit higher spectral entropy, and Primates, which exhibit lower spectral entropy, are distinguishable based on their curves. In contrast, the spectral entropy curves for Carnivora, Perissodactyla, Cetartiodactyla, and Rodentia are closer together in weighted networks.

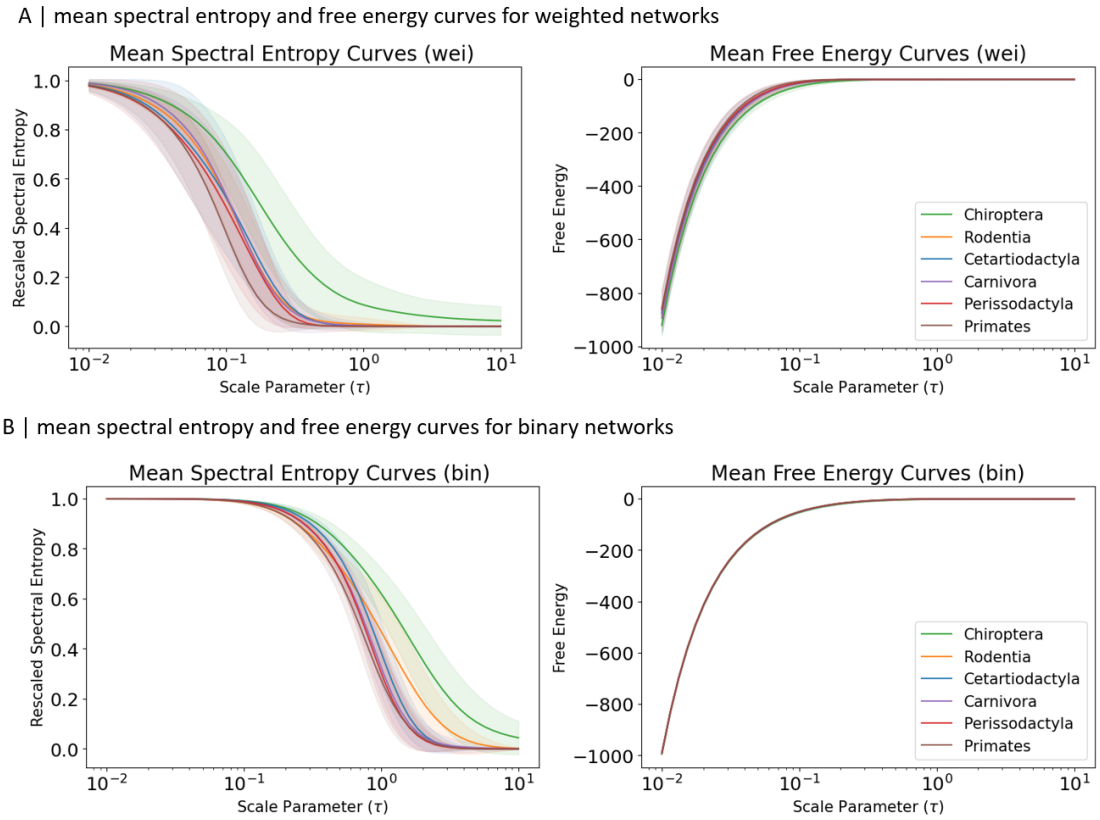

Figure 3.5: Mean spectral entropies and free energy profiles corresponding to the phylogenetic tree based on taxonomy. Part A presents the mean spectral entropy and free energy for weighted networks, while Part B shows these curves for binarized networks. The curves represent the averaged spectral entropies and free energy across different scale parameters τ for each order of species. Standard deviations are shown at each scale, providing insight into the variability across species connectomes within the same order.

3.2.4 Inter- and Intra-Order Similarities in the Order-Based Distance Matrix

One of the objectives we are pursuing is to demonstrate that connectomes exhibit greater similarity with connectomes from the same order based on the phylogenetic

tree. In the paper by Suarez et al. [51], this similarity was shown using topological features. We are using methods based on statistical mechanics that integrate the analysis from a functional approach.

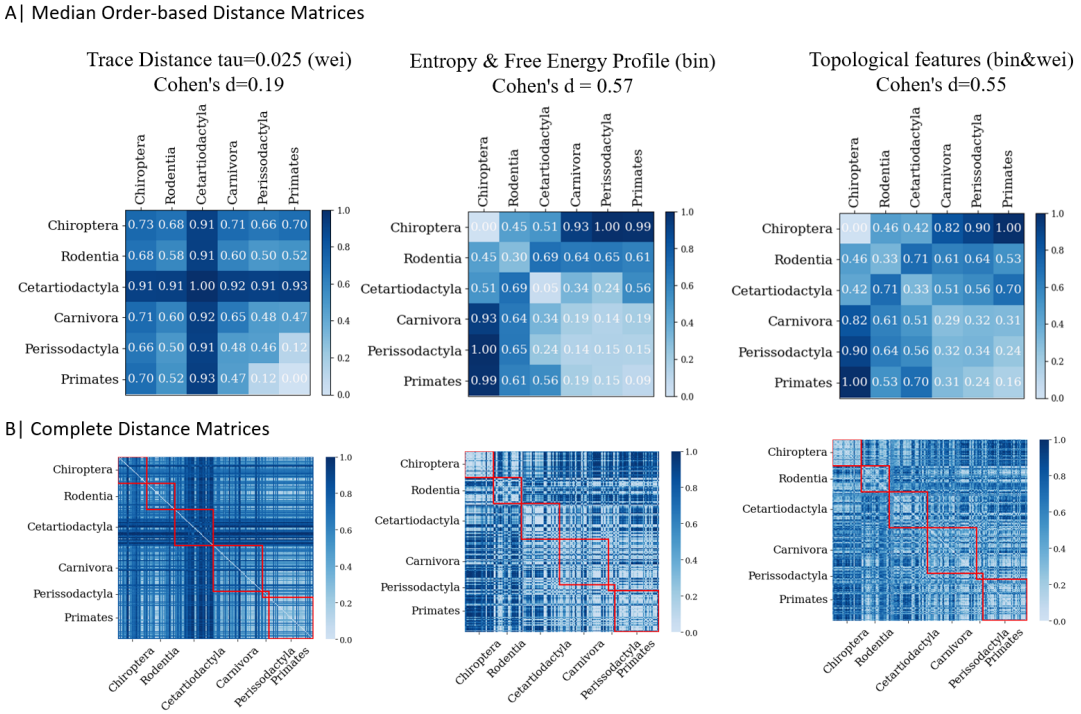

Figure 3.6: Order-Based Distance Matrices. Part A shows the mean distances obtained through the sampling method, along with Cohen's d, indicating inter-order similarity. In Part B, the complete distance matrices are displayed, with red squares highlighting different orders.

To calculate inter- and intra-order similarity, we employ Cohen's D, a measure of effect size that quantifies the difference between two means in terms of standard deviation. It is commonly used to compare the means of two groups and is defined as:

$$d = \frac{\overline{X_1} - \overline{X_2}}{s_p} \tag{3.1}$$

where $\overline{X_1}$ and $\overline{X_2}$ are the means of the two groups being compared, and s_p is the pooled standard deviation. The pooled standard deviation is calculated as:

$$s_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \tag{3.2}$$

Here, n_1 and n_2 are the sample sizes of the two groups, and s_1^2 and s_2^2 are the variances of the two groups.

To assess the overall inter- and intra-order similarity, we calculate the mean of all pairwise Cohen's D values across different orders. three of the best possible scores for each method (Trace distance, entropy-free energy profile, and topological features) are reported in Figure 3.6. This provides a comprehensive measure of how similar connectomes within the same order are compared to those from different orders.

However, because some species have multiple scans, this could bias the distribution of intra- and inter-order distances, which could be dominated by those species with a large number of replicas. To account for this potential bias, we randomly sample a single connectome per species and calculate inter-species distances. We repeat this procedure iteratively 10,000 times. The reported intra- and inter-order distance distributions correspond to the average distances across these iterations.

3.2.5 Clustering the Connectomes of the MAMI Dataset Using Functional and Topological Features

In machine learning, clustering is a type of unsupervised learning technique that involves grouping a set of objects or data points into clusters based on their similarities. The goal of clustering is to ensure that objects within the same cluster are more similar to each other than to those in other clusters. We applied a clustering algorithm to find a taxonomy, based on the mammalian connectomes. For this study, we explored various clustering methods such as hierarchical, K-Means, and DBSCAN and we found that the best algorithm firs to our task is Agglomerative clustering. This clustering algorithm is a hierarchical method, that was chosen based on its performance. This approach starts with individual data points and merges the two most similar clusters iteratively until all points are unified into a single cluster or a predefined number of clusters is achieved. Among the linkage criteria available for agglomerative clustering, Ward's linkage was employed in this study.

Ward's linkage minimizes the total within-cluster variance. It merges clusters by evaluating the increase in the sum of squared deviations from the cluster mean that would result from merging each pair of clusters. We utilized the `scikit-learn` library [44] for applying this algorithm. We used UMAP instead of MDS because UMAP (Uniform Manifold Approximation and Projection) provides a more accurate representation of high-dimensional data in lower dimensions, preserving both local and global structures more effectively than MDS. Unlike MDS, which focuses on preserving pairwise distances, UMAP captures more complex relationships between data points, leading to better clustering and visualization results. Clustering was performed on the projected features or distance matrices in a 25-dimensional space, with visualization done in 2 dimensions. Figure 3.7 shows the results of the top three clustering methods, including the corresponding clustered labels and their order.

To compare the clustering obtained from connectomes to phylogenetic trees, we used two metrics: Normalized Mutual Information (NMI) and Adjusted Mutual Information (AMI) [54].

Normalized Mutual Information (NMI): NMI is a normalization of the Mutual Information (MI), which measures the similarity between two clustering results by comparing the mutual information to the average entropy of the clusters. It is defined as:

where $H(U)$ and $H(V)$ are the Shannon entropies of the clusterings U and V, respectively, and $I(U;V)$ represents the mutual information between the two clusterings. calculated as:

$$I(U;V) = \sum_{u \in U} \sum_{v \in V} P(u,v) \log \left(\frac{P(u,v)}{P(u)P(v)} \right) \tag{3.3}$$

NMI is normalized between 0 and 1, where 1 indicates perfect agreement between

the two clusterings, and 0 indicates no mutual information, implying complete independence of the clusters.

Adjusted Mutual Information (AMI): AMI modifies the mutual information score to account for chance. It is particularly useful in ensuring that the expected value of AMI approaches zero when the clustering results are similar to those expected under random clustering conditions, regardless of the number of clusters involved. AMI is defined as:

$$\mathrm{AMI}(U,V) = \frac{I(U;V) - \mathbb{E}[I(U;V)]}{\frac{H(U)+H(V)}{2} - \mathbb{E}[I(U;V)]} \tag{3.4}$$

where $\mathbb{E}[I(U;V)]$ represents the expected mutual information calculated under the assumption that the clustering assignments are independent but constrained to the observed data margins.

3.2.6 Results for All Possible Combinations of Methods

The options for projection, combined with binary and weighted graphs, present a large number of possible configurations. We explored all of these combinations to identify the most effective configuration for each approach: Trace Distance, entropy-free energy profile, and topological features. The clustering method used was Agglomerative Clustering with Ward's linkage, set to a fixed number of six clusters, corresponding to the number of orders we analyzed. MDS and UMAP were employed as dimensionality reduction techniques to enhance clustering performance. We analyzed two types of networks: binary and weighted. Additionally, it is possible to combine entropy and free energy profiles. All the scores for these combinations are reported in Table 3.1.

The most aligned clustering for the topological method was achieved using both binary and weighted features, with an NMI of 0.3. For the entropy and free energy profiles, using both for binary networks resulted in an NMI of 0.28. For the Trace distance at $\tau = 0.025$ for weighted networks, using UMAP and MDS yielded an NMI of 0.18. The results for the entropy profile and the topological features are quite similar and from the ordered-based distance matrices shown in Figure 3.6, we can see the alignment of these distance matrices. The orders that are similar to each other which are Primates, Rodentia, and Perissodactyla in the topological approach are also similar form the entropy and free energy profile method which is a functional analysis.

3.3 Summary

In this chapter, we applied the functional analysis described in previous chapters and examined the density matrices using test networks generated by BA and ER models. The results demonstrate the sensitivity of the functional analysis to network connectivity and its responsiveness to different parameters of preferential attachment in the BA model. Alongside topological features, we explored the relationship between connectomic-based taxonomy and the phylogenetic tree using functional analysis. Our comparison of methods revealed similarities within species of the same order and differences between species of different orders by comparing their connectomes.

Table 3.1: Comparison of Method Combinations Based on NMI and AMI

Method	**NMI**	**AMI**
Topological wei directly	0.27	0.24
Topological wei UMAP	0.28	0.25
Topological wei MDS	0.26	0.22
Topological bin directly	0.17	0.14
Topological bin UMAP	0.19	0.16
Topological bin MDS	0.20	0.16
Topological bin-wei directly	0.30	0.27
Topological bin-wei UMAP	0.30	0.27
Topological bin-wei MDS	0.30	0.27
Trace Distance wei tau=0.025 UMAP	0.18	0.14
Trace Distance wei tau=0.025 MDS	0.18	0.14
Trace Distance bin tau=0.025 MDS	0.05	0.00
Trace Distance bin tau=0.025 UMAP	0.02	0.00
Entropy & free energy profile wei UMAP	0.19	0.15
Entropy & free energy profile wei directly	0.20	0.16
Entropy & free energy profile wei MDS	0.16	0.12
Entropy & free energy profile bin UMAP	0.28	0.25
Entropy & free energy profile bin directly	0.25	0.22
Entropy & free energy profile bin MDS	0.20	0.16
Entropy & free energy profile wei-bin UMAP	0.23	0.20
Entropy & free energy profile wei-bin directly	0.23	0.20
Entropy & free energy profile wei-bin MDS	0.24	0.20
Entropy profile wei UMAP	0.20	0.16
Entropy profile wei directly	0.15	0.11
Entropy profile wei MDS	0.11	0.15
Entropy profile bin UMAP	0.27	0.23
Entropy profile bin directly	0.21	0.17
Entropy profile bin MDS	0.27	0.23
Free energy profile wei UMAP	0.21	0.18
Free energy profile wei directly	0.18	0.14
Free energy profile wei MDS	0.18	0.14
Free energy profile bin UMAP	0.20	0.17
Free energy profile bin directly	0.20	0.17
Free energy profile bin MDS	0.20	0.17

The results from both analyses corroborate each other, suggesting that functional information is embedded within topological features. In the next chapter, we will use entropy profiles from functional analysis and optimization algorithms to generate networks using various generative models that closely resemble real-world networks, such as connectomes.

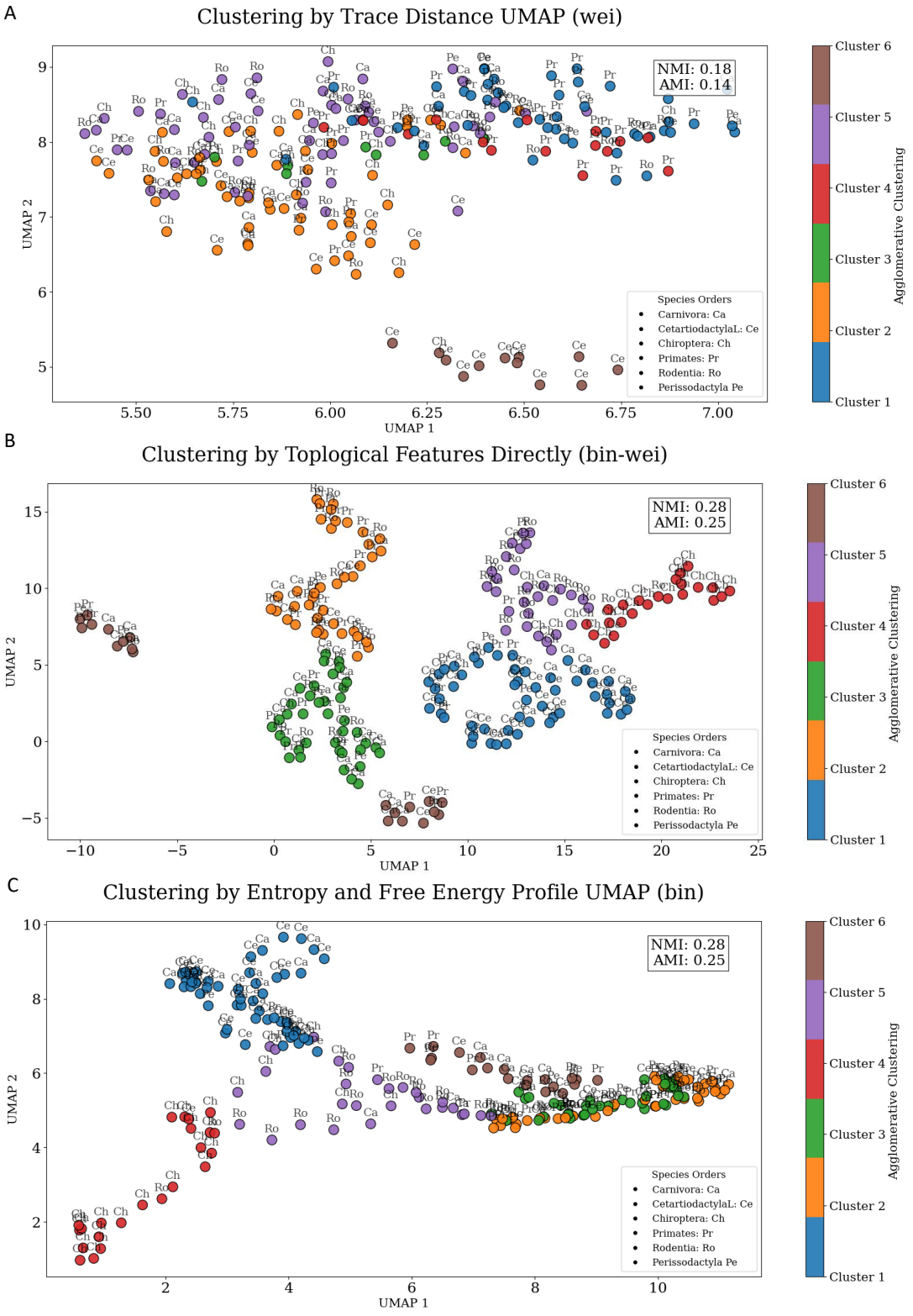

Figure 3.7: Clustering of Connectomes using Functional and Topological Features. (A) Clustering based on distance matrices at $\tau = 0.025$ from the trace distance; (B) Clustering of entropy and free energy profile features; (C) Clustering based on topological features. Colors indicate clusters identified by agglomerative clustering, with text labels denoting phylogenetic tree orders. NMI and AMI scores are provided to compare clustering with phylogenetic orders.

Chapter 4

Computational Modeling of Biological Networks

In this chapter, we focus on computational network modeling with the goal of optimizing procedures for aligning theoretical models with real-world biological networks, particularly complex networks like the Mammalian Connectome. The work begins by assessing the distance between networks using entropy profile vectors. Optimization algorithms, such as Simulated Annealing and Particle Swarm, are then applied to minimize this distance, effectively capturing information pathways across multiple scales. Different parameters within generative models, including the Erdős-Rényi model, Barabási-Albert model, Watts-Strogatz model, and the Stochastic Block Model (SBM), are explored by optimization algorithms to generate networks that closely resemble the target network. Jensen-Shannon Divergence (JSD) is subsequently employed to compare the generated networks with the target network based on their density matrices at various scales.

4.1 Optimization Algorithms

To explore different parameters of generative models and find the most similar network to the target, we need to employ optimization algorithms. These algorithms are designed to find the best set of parameters that minimize (or maximize) an objective function. Among the many available methods, we chose Particle Swarm Optimization (PSO) and Simulated Annealing (SA), two widely used algorithms with distinct advantages. PSO is inspired by the social behavior of birds flocking or fish schooling, where a group of candidate solutions, called particles, move through the parameter space by updating their positions and velocities based on both individual and collective experiences. This method is particularly effective for continuous optimization problems where the solution space is complex and multi-dimensional. This optimization algorithm will be more useful when we are using generative models with more parameters. On the other hand, SA is inspired by the annealing process in metallurgy and employs a probabilistic technique to escape local optima by accepting worse solutions with a certain probability, which decreases over time. SA is especially useful in discrete optimization problems or when the solution space is highly irregular. In the following subsections, we will thoroughly explain these two optimization algorithms.

4.1.1 Simulated Annealing

Simulated Annealing (SA) is a stochastic optimization method used for finding near-optimal solutions for an optimization problem. The algorithm is useful when looking for a global solution in a search space that may contain local optimas which other methods may end up with. This algorithm is named after the process of annealing used in the metallurgical industry. a process whereby a material is heated and then gradually cooled in a manner that will eliminate the defects in its structure.

The key control parameter of SA is the temperature T, which controls the limitation of the exploration area in the system. It controls the acceptance of worse solutions as the algorithm progresses. The probability of accepting a worse solution is governed by the Metropolis criterion:

$$P(\Delta E) = \exp\left(-\frac{\Delta E}{k_B T}\right) \tag{4.1}$$

where ΔE is the difference in the objective function between the current solution and a neighboring solution, and T is the current temperature and we assume $k_B = 1$. When T decreases, the probability of accepting worse solutions reduces and it allows the algorithm to reach to an optimal solution during iterations.[33].

The general steps of the SA algorithm are as follows:

Algorithm 1 Simulated Annealing

```
Initialize: T = T_0, current solution s, best solution s*
while stopping criterion not met do
    Generate a neighbor solution s'
    Compute ΔE = f(s') − f(s)
    if ΔE < 0 or exp(−ΔE/T) > random(0, 1) then
        s ← s'
    end if
    if f(s) < f(s*) then
        s* ← s
    end if
    Update T according to the cooling schedule
end while
return s*
```

f is the objective function that we want to minimize or maximize it. The performance of SA heavily depends on the cooling schedule and the definition of the neighborhood structure. The cooling schedule typically follows an exponential decay, $T = T_0 \times \alpha^k$, where α is a constant ($0 < \alpha < 1$), and k is the iteration number [1].

4.1.2 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is an iterative optimization algorithm used to search for the optimal solution to a problem by improving a candidate solution with respect to a particular metric. It is inspired by bird migration or fish schooling. The algorithm operates by creating a group of particles that represent a set of potential

solutions, and then shifting these particles within the search space according to simple mathematical formulas that involve each particle's position and velocity.

Each particle's movement is updated according to its own best-known position and the best-known positions of the swarm in general, searching for the global optimum. The position $\mathbf{x}_i$ and velocity $\mathbf{v}_i$ of each particle i are updated according to the following equations:

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1 r_1(\mathbf{p}_i - \mathbf{x}_i(t)) + c_2 r_2(\mathbf{g} - \mathbf{x}_i(t))$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1)$$

where: $\mathbf{v}_i(t)$ is the velocity of particle i at iteration t, $\mathbf{x}_i(t)$ is the position of particle i at iteration t, $\mathbf{p}_i$ is the best-known position of particle i, $\mathbf{g}$ is the global best-known position among all particles, w is the inertia weight, c_1 and c_2 are cognitive and social coefficients, respectively, r_1 and r_2 are random values uniformly distributed in $[0, 1]$.

The PSO algorithm is as follows:

Algorithm 2 Particle Swarm Optimization

```
Initialize the swarm x_i and velocities v_i for all particles i
while stopping criterion not met do
    for each particle i do
        Update velocity v_i(t+1) using the velocity update equation
        Update position x_i(t+1)
        Evaluate the fitness of x_i(t+1)
        if x_i(t+1) is better than p_i then
            p_i ← x_i(t+1)
        end if
    end for
    Update the global best position g
end while
return g
```

PSO has been widely used due to its simplicity and effectiveness in finding near-optimal solutions for complex optimization problems. Its performance can be significantly influenced by the choice of parameters, such as the inertia weight w, and the cognitive and social coefficients c_1 and c_2 [31, 12].

4.2 Network Fitting by Minimizing Entropy Profiles through Optimizing Synthetic Networks

We present a pipeline for fitting generative models to real networks based on the entropy profile distances between them. This is done using optimization methods such as simulated annealing and particle swarm optimization. These methods help find the best parameters for synthetic networks that offer the closest fit to the empirical data.

The entropy profile allows us to observe the diversity of pathways at different scales. Calculating the entropy profile is computationally efficient. This efficiency is crucial, especially for optimization algorithms where multiple calculations are needed to obtain the final result. As explained in Section 3.2.3, the entropy profile can be calculated from the eigenvalues of the Laplacian matrix of networks. From this, we can obtain entropy at various scales, known as spectral entropy. The entropy profile is a vector of entropies across different scales.

Using optimization methods, the goal is to minimize the distance between the entropy profiles of the target network $S(m)$ and the synthetic network $S(v)$. The aim is to find the closest possible match by exploring various parameters of the synthetic networks obtained by generative models. The distance is defined as:

$$\text{distance}(S(m), S(v)) = \sqrt{\sum_{i=1}^{N} \left(S(m)_i - S(v)_i\right)^2} \tag{4.2}$$

By optimizing the parameter θ, we find the best fit of the synthetic network:

$$\text{Objective}(\theta) = \text{distance}(S(m), S(v(\theta))) \quad , \quad \theta = [\theta_1, \theta_2, \ldots, \theta_k] \tag{4.3}$$

$$\theta^* = \arg\min_{\theta} \text{Objective}(\theta) \tag{4.4}$$

To examine the fitting algorithm, we set the target network as an ER with $p = 0.4$, and 200 nodes, and tried to find the closest p by the method which has been explained. This involved employing a Simulated Annealing optimization algorithm to explore various p parameters and find the most suitable fit from the ER networks. In Figure 4.1, we can see the exploration of the Simulated Annealing optimization algorithm to minimize the objective function, which is the entropy profile distance.

In the next step, the optimization was repeated 50 times. A close solution to the target network can be found by the algorithm. It was found that $p = 0.39$ by minimizing the objective function, which is the entropy profile distance, and the minimum entropy distance found was 0.05. In Figure 4.2, we can see the distribution of the optimization results and the mean of them to compare to the p which has been selected for the target network. The histogram demonstrates that solutions to the simplest problem with small errors can be found by the Simulated Annealing optimization method, along with the selected objective function. Moreover, similar results were obtained with the Particle Swarm Optimization algorithm, and the distribution has been shown in Figure 4.3.

In the next step, the target network was changed to Barabasi-Albert with $m = 20$ and 100 nodes, resulting in 1600 edges. Analytically, the best fit of an ER network should only take care of the number of edges and should be an ER with $p = 0.32$. However, since the objective function minimizes the difference of entropy in different scales, the result will be different from the analytic solution. This method will find the network that has the most similarity from short to long scales. The results are shown in Figure 4.4. A similar result was obtained for this experiment ($p = 0.278$) using the Particle Swarm optimization algorithm, with less standard deviation (Figure 4.5).

The best fit for this BA network, achieved by minimizing the entropy profiles and ER model, is an ER network with $p = 0.282$. Since only the number of edges

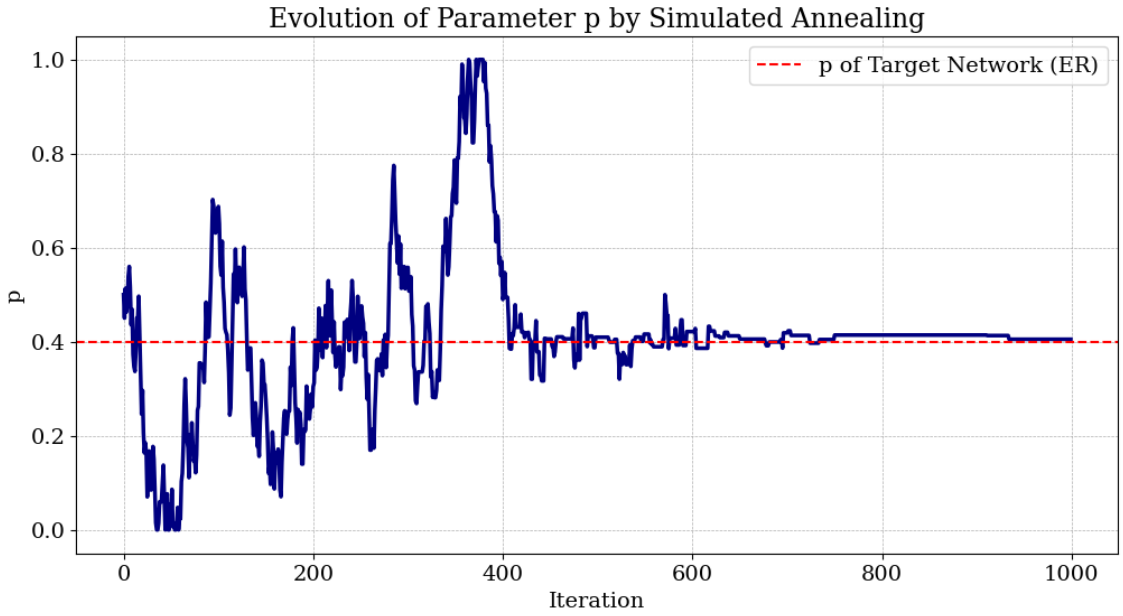

Figure 4.1: Exploring the parameter p in the ER model using Simulated Annealing. The figure illustrates the exploration process of the Simulated Annealing optimization algorithm applied to the ER model over 1000 iterations to find a target network which is ER with $p = 0.4$, by minimizing the entropy profile distance.

is changed with the ER model, from the analytic solution, the most similar ER network should be an ER with $p = 0.32$. However, since this method takes care of the diversity of pathways at different scales, the result is different. This ability could be useful when fitting models with more complexity to retain more information from the target network. It is also possible to find networks that have similarities in a selected range of τ scale parameters to personalize the fitting if similarity in selective scales is needed. Nevertheless, it is important to note that from a purely structural perspective, this method may be inconclusive.

4.3 Comparison of Different Models Using Jensen-Shannon Divergence

To compare the performance of methods, networks will be generated in different ways by different models, and the similarity of each model's output to the target network will be measured. First, the target will be set with a network generated by generative models with known parameters, and in the end, the method will be applied to a real Mammalian connectome from the MAMI dataset. The configuration model will be used to generate networks from the degree sequence. Additionally, the ERGM package[25] in R can be utilized to generate graphs similar to the target network using the Exponential Random Graph model. Furthermore, networks that are close to the real one according to their spectral entropy are obtained from the fitted models. To assess the information learned about the empirical network through the use of the models, the Jensen-Shannon divergence between synthetic networks and the empirical one is calculated, which refers to the amount of information lost using the models (in bits).

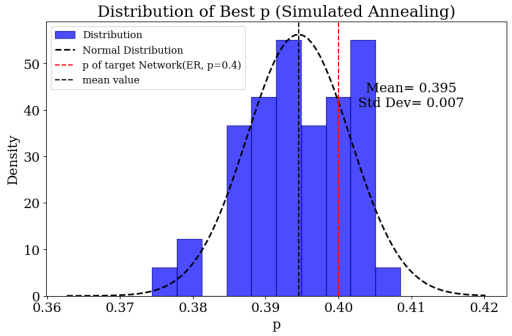

Figure 4.2: Distribution of results obtained by running the Simulated Annealing optimization algorithm 50 times for target network ER with p=0.4.

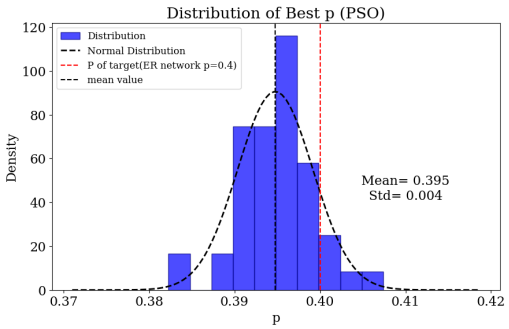

Figure 4.3: Distribution of results obtained by running the Particle Swarm Optimization algorithm 50 times for target network ER with p=0.4.

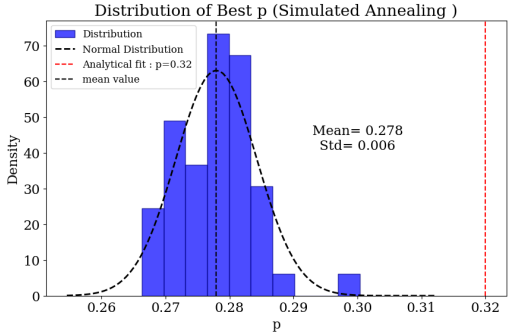

Figure 4.4: Distribution of results obtained by running the Simulated Annealing optimization algorithm 50 times for BA with $m = 20$.

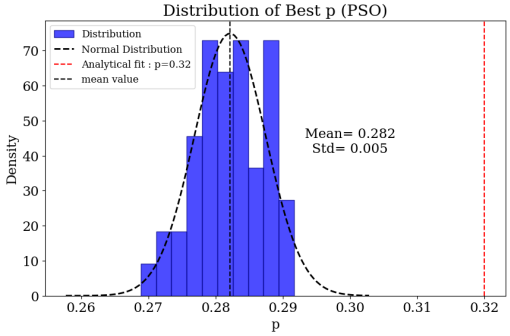

Figure 4.5: Distribution of results obtained by running the Particle Swarm Optimization algorithm 50 times for target network BA with $m = 20$.

The target network was set as a Barabasi Albert with $m = 10$ and 200 nodes. Different models including Barabasi Albert, Erdos Reyni, Watts-Strogats model, and SBM with two communities were utilized to find the most similar network using both Particle Swarm and Simulated Annealing optimizations. The Jensen-Shannon Divergence (JSD) of their density matrices in different scales was calculated to observe the difference between the generated networks and the target network, which is a Barabasi-Albert in this case. For each model, the optimization was run 50 times to obtain the minimum entropy profile distance. 40 networks were generated for each model, and the JSD was calculated for them. The curves in Figure 4.6 represent the mean of the Jensen-Shannon Divergence (JSD) for all networks generated from each model. The standard deviation is also depicted with shadows to visualize the distribution of generated networks, which results from the randomness inherent in each generative model. The efficiency of each model was evaluated by integrating the area under the curve to determine which model has less difference with the target network. The results show that the configuration model and also since we chose the target network Barabasi-Albert, using the method by the Barabasi-Albert model has a low JSD as well.

Moreover, to verify the results with another model, we set the target model as a

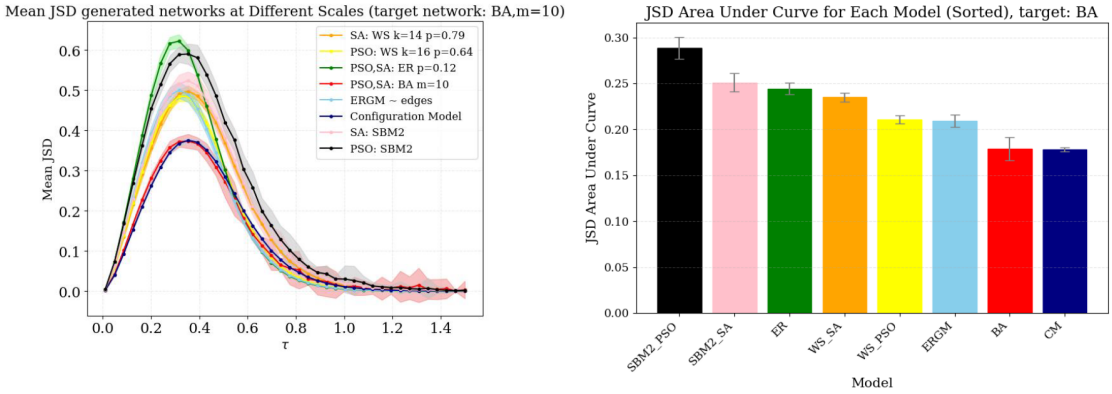

Figure 4.6: JSD comparison of the fitted networks on the target network Barabasi-Albert. On the left: Mean Jensen-Shannon Divergence (JSD) between density matrices of generated networks from different models and the target network Barabasi-Albert with $m = 10$ at various scales. On the right: the sorted integrated area under the curves for each model. Lower JSD values indicate a better fit, meaning the fitted networks are more similar to the target network.

Watts-Strogatz network, which is a small-world network with $k = 16$ and $p = 0.65$. The results show that the Configuration model generates networks with a lower JSD. Additionally, it is expected that by choosing the Watts-Strogatz model and using the optimization algorithm, we will obtain a low JSD (see Figure 4.7). This indicates that by selecting the appropriate model for the data, we can generate similar networks and potentially discover the underlying rules of the empirical networks.

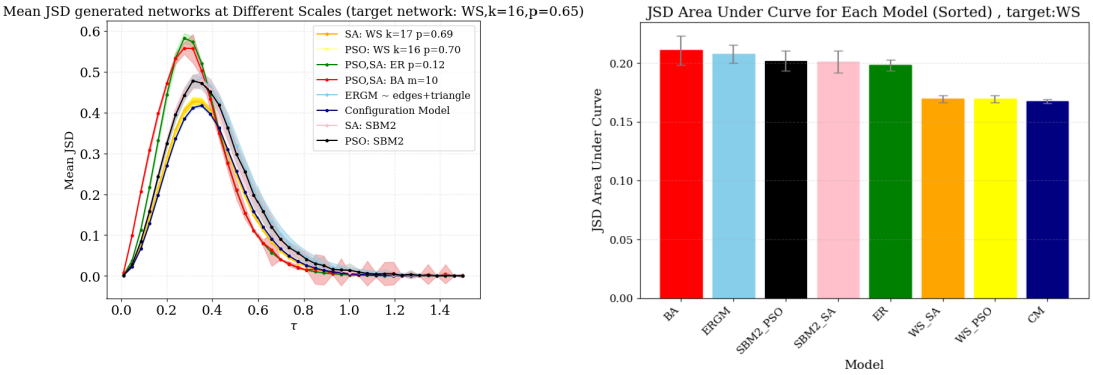

Figure 4.7: JSD comparison of the fitted networks on the target network Watts-Strogatz. On the left: MeanJensen-Shannon Divergence (JSD) between density matrices of generated networks from different models and the target network Watts-Strogatz with $k = 16$ and $p = 0.65$ at various scales. On the right: the sorted integrated area under the curves for each model. Lower JSD values indicate a better fit, meaning the fitted networks are more similar to the target network.

In the end, we set the target network as the real Cat connectome obtained from the MAMI dataset to evaluate the performance of the method in real-world cases. The results show that the Barabasi-Albert model generates the most similar

networks with the parameters obtained by optimizations (see Figure 4.8). However, the Configuration model also generates similar networks with a small JSD and a small standard deviation.

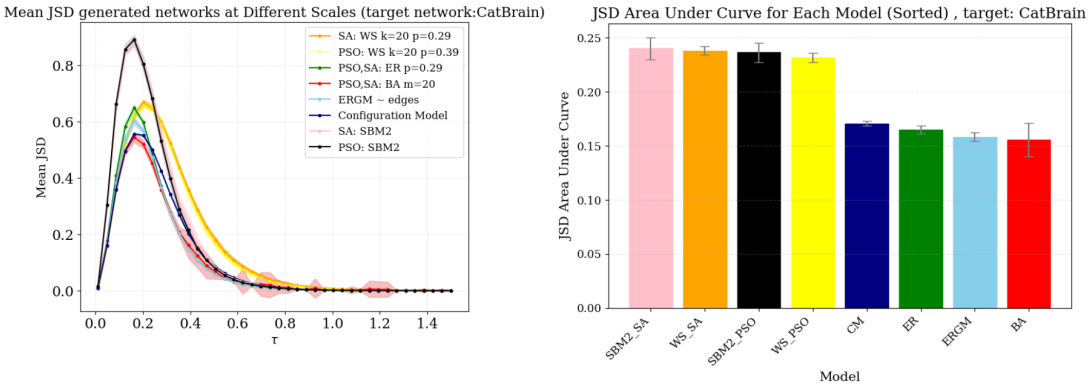

Figure 4.8: JSD comparison of the fitted networks on the target network Cat connectome. On the left: Mean Jensen-Shannon Divergence (JSD) between density matrices of generated networks from different models and the target network Cat connectome at various scales. On the right: the sorted integrated area under the curves for each model. Lower JSD values indicate a better fit, meaning the fitted networks are more similar to the target network.

4.4 Summary

This section comprehensively explored the effectiveness of various generative models, along with optimization algorithms, in generating networks similar to complex biological networks based on their density matrices. Through experimentation, we evaluated the capability of these models to fit biological networks by minimizing entropy profiles and comparing their performance with the Configuration model and ERGM. By creating target networks using generative models, such as Barabási-Albert, we explored model parameters using optimization algorithms. This allowed us to obtain parameters close to those of the target model when using the same generative model, thereby demonstrating the effectiveness of our method and indicating that the entropy profile contains most of the network's information.

Assessment via JSD revealed that the Configuration model and models closely resembling the target network exhibited the most promising outcomes, with the Configuration model's preservation of node order providing an edge, particularly given JSD's sensitivity to node order when the target network is generated via a generative model. Furthermore, our examination of a real mammalian connectome highlighted the Barabási-Albert model, optimized through entropy profile distance, as yielding superior results. This suggests a propensity towards power-law distribution in real networks.

Chapter 5

Conclusions

In this work, we utilized advanced tools from statistical mechanics, specifically the density matrix and spectral entropy, an information-theoretic measure, to perform functional analysis and compare complex networks. This approach allowed for a more comprehensive comparison of complex networks, enabling analysis across a range of scales, from short to long, with each scale capturing various network properties. By introducing entropy and free energy profiles, we developed a more general functional tool that accounts for all scales at the same time.

Using these tools, we conducted a generalized analysis of the mammalian connectome from the MAMI dataset to identify differences in distance matrices and perform clustering. Notably, our clustering results exhibited some similarities with the genetic-based taxonomy of species. By comparing the functional approach to the topological one, we found consistent results, indicating that the information encoded in the entropy profiles and density matrices at various scales aligns with local and global features from network science and graph theory, such as clustering, betweenness, and assortativity.

We performed an NMI-based comparison using different combinations of methods and identified the best approaches within each method. In terms of topological features, using binary and weighted features together achieved an NMI of 0.3. The best result for the trace distance of density matrices was found at $\tau = 0.025$ with an NMI of 0.18. Additionally, the entropy and free energy profile projection using UMAP achieved an NMI of 0.28, which was the most similar clustering to the phylogenetic tree taxonomy. The distance matrices derived from entropy profiles, free energy (Cohen's D = 0.57), and topological features (Cohen's D = 0.55) demonstrated the greatest intra-order similarity and inter-order differences. The similarity of the distance matrices suggests that there is mutual information embedded in the entropy and free energy profiles across different scales and in the topological features.

Finally, we introduced a method that incorporates spectral entropies and optimization techniques to fit generative models to empirical networks. By applying this method in some experiments, we fixed the target network generated by a specific generative model. Our fitting approach, using the same generative model as the target network, successfully generated networks most similar to the target. Additionally, the configuration model also performed well in generating similar networks. By setting the cat connectome as the target network, our method suggested the possibility of approximating it with a Barabási-Albert network, albeit with some deviation. While further experimentation with diverse models and parameters is necessary for

validation, our research highlights the importance of entropy profiles in capturing pathway diversities across different scales in biological network modeling.

Overall, this work demonstrates the effectiveness and utility of network density matrices in the comparison and feature extraction of networks. By introducing a general formalism for analyzing networks across different scales, this approach offers a controllable and comprehensive method for extracting information from complex networks—information that cannot be fully captured by traditional topological measures alone.

Bibliography

[1] E. H. L. Aarts and J. H. M. Korst. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Chichester, England: John Wiley & Sons, 1989. ISBN: 9780471921462.

[2] Alessia Amelio and Clara Pizzuti. "Correction for closeness: Adjusting normalized mutual information measure for clustering comparison". In: *Computational Intelligence* 33.3 (2017), pp. 579–601.

[3] Yaniv Assaf et al. "The Connected Brain: Multi-Level Models of the Impact of the Connectome on Disease". In: *Nature Reviews Neuroscience* 21 (2020), pp. 303–318. DOI: 10.1038/s41583-020-0287-8.

[4] F.A.C. Azevedo et al. "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain". In: *Journal of Comparative Neurology* 513 (2009), pp. 532–541. DOI: 10.1002/cne.21974.

[5] Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks". In: *Science* 286.5439 (1999), pp. 509–512.

[6] Alain Barrat et al. "The architecture of complex weighted networks". In: *Proceedings of the national academy of sciences* 101.11 (2004), pp. 3747–3752.

[7] Peter J. Basser et al. "In vivo fiber tractography using DT-MRI data". In: *Magnetic Resonance in Medicine* 44 (2000), pp. 625–632. DOI: 10.1002/1522-2594(200010)44:4<625::AID-MRM17>3.0.CO;2-O.

[8] Jacob Biamonte and Ville Bergholm. "Tensor networks in a nutshell". In: *Nature Physics* 13.6 (2017), pp. 623–630. DOI: 10.1038/nphys3966.

[9] Ulrik Brandes. "A faster algorithm for betweenness centrality". In: *Journal of Mathematical Sociology* 25.2 (2001), pp. 163–177.

[10] Ed Bullmore and Olaf Sporns. "Complex brain networks: graph theoretical analysis of structural and functional systems". In: *Nature Reviews Neuroscience* 10 (2009), pp. 186–198. DOI: 10.1038/nrn2575.

[11] Fan RK Chung. "Spectral graph theory". In: *CBMS Regional Conference Series in Mathematics* 92 (1997).

[12] Maurice Clerc. *Particle Swarm Optimization*. London, UK: Wiley, 2002. ISBN: 978-1-84919-121-8.

[13] Manlio De Domenico and Jacob Biamonte. "Spectral entropies as information-theoretic tools for complex network comparison". In: *Physical Review X* 6.4 (2016), p. 041062.

[14] Manlio De Domenico et al. "Complex Networks: From Classical to Quantum". In: *Physics Reports* 684 (2017), pp. 1–50. DOI: 10.1016/j.physrep.2017.05.002.

[15] Manlio De Domenico et al. "Mathematical formulation of multilayer networks". In: *Physical Review X* 3.4 (2013), p. 041022.

[16] Frédéric Delsuc, Henner Brinkmann, and Hervé Philippe. "Phylogenomics and the reconstruction of the tree of life". In: *Nature Reviews Genetics* 6.5 (2005), pp. 361–375.

[17] P. Erdős and A. Rényi. "On random graphs". In: *Publicationes Mathematicae Debrecen* 6 (1959), pp. 290–297.

[18] Ernesto Estrada. *Structure and dynamics of complex networks*. Oxford University Press, 2012.

[19] Ernesto Estrada. "The communicability distance in graphs". In: *Linear Algebra and its Applications* 436.11 (2012), pp. 4317–4328. DOI: 10.1016/j.laa.2012.01.002.

[20] European Bioinformatics Institute. *European Bioinformatics Institute (EMBL-EBI)*. https://www.ebi.ac.uk/. [Accessed August 16, 2024]. 2024.

[21] U. Fano. "Description of States in Quantum Mechanics by Density Matrix and Operator Techniques". In: *Rev. Mod. Phys.* 29 (1 Jan. 1957), pp. 74–93. DOI: 10.1103/RevModPhys.29.74.

[22] Linton C Freeman. "A set of measures of centrality based on betweenness". In: *Sociometry* (1977), pp. 35–41.

[23] G.H. Glover. "Overview of functional magnetic resonance imaging". In: *Neurosurgical Clinics of North America* 22.2 (Apr. 2011), pp. 133–139. DOI: 10.1016/j.nec.2010.11.001.

[24] Aric Hagberg, Pieter Swart, and Daniel S Chult. *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[25] Mark S. Handcock et al. *ergm: Fit, Simulate and Diagnose Exponential-Family Models for Networks*. R package version 4.6.0. The Statnet Project. 2023. URL: https://CRAN.R-project.org/package=ergm.

[26] Suzana Herculano-Houzel. "The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost". In: *Proceedings of the National Academy of Sciences* 109 (2012), pp. 10661–10668. DOI: 10.1073/pnas.1201895109.

[27] David R Hunter et al. "ergm: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks". In: *Journal of Statistical Software* 24.3 (2008), nihpa54860. DOI: 10.18637/jss.v024.i03.

[28] Edwin T. Jaynes. "Information Theory and Statistical Mechanics". In: *Physical Review Series II* 106.4 (1957), pp. 620–630. DOI: 10.1103/PhysRev.106.620.

[29] Edwin T. Jaynes. "Information Theory and Statistical Mechanics II". In: *Physical Review Series II* 108.2 (1957), pp. 171–190. DOI: 10.1103/PhysRev.108.171.

[30] Eric R. Kandel et al. *Principles of Neural Science*. 6th edition. McGraw-Hill Education, 2021.

[31] James Kennedy and Russell Eberhart. "Particle Swarm Optimization". In: *Proceedings of the IEEE International Conference on Neural Networks*. IEEE. 1995, pp. 1942–1948. DOI: 10.1109/ICNN.1995.488968.

[32] Shiva Kintali. "Betweenness centrality: Algorithms and lower bounds". In: *Proceedings of the 2nd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems (PODS)*. 2008, pp. 122–123.

[33] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. "Optimization by Simulated Annealing". In: *Science* 220.4598 (1983), pp. 671–680. DOI: 10.1126/science.220.4598.671.

[34] Vito Latora and Massimo Marchiori. "Efficient behavior of small-world networks". In: *Physical review letters* 87.19 (2001), p. 198701.

[35] I X Y Leung et al. "Assortativity, cores, and clustering in social networks". In: *Physica A: Statistical Mechanics and its Applications* 378.2 (2007), pp. 591–604.

[36] David Meunier, Renaud Lambiotte, and Edward T. Bullmore. "Modular and hierarchically modular organization of brain networks". In: *Frontiers in Neuroscience* 4 (2010), p. 200. DOI: 10.3389/fnins.2010.00200.

[37] Michael Molloy and Bruce Reed. "A critical point for random graphs with a given degree sequence". In: *Random structures & algorithms* 6.2-3 (1995), pp. 161–180.

[38] Mark Newman. *Networks*. Oxford University Press, 2018.

[39] Mark EJ Newman. "Assortative mixing in networks". In: *Physical review letters* 89.20 (2002), p. 208701.

[40] Mark EJ Newman. "The structure and function of complex networks". In: *SIAM review* 45.2 (2003), pp. 167–256.

[41] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[42] Jukka-Pekka Onnela et al. "Intensity and coherence of motifs in weighted complex networks". In: *Physical Review E* 71.6 (2005), p. 065103.

[43] D. Papo and J.M. Buldú. "Does the brain behave like a (complex) network? I. Dynamics". In: *Physics of Life Reviews* 48 (2024), pp. 47–98. ISSN: 1571-0645. DOI: 10.1016/j.plrev.2023.12.006. URL: https://www.sciencedirect.com/science/article/pii/S1571064523002105.

[44] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830.

[45] Garry Robins et al. "An introduction to exponential random graph (p*) models for social networks". In: *Social Networks* 29.2 (2007), pp. 173–191.

[46] Mikail Rubinov and Olaf Sporns. "Complex network measures of brain connectivity: uses and interpretations". In: *Neuroimage* 52.3 (2010), pp. 1059–1069.

[47] Claude E. Shannon. "A mathematical theory of communication". In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423.

[48] Tom AB Snijders. "Markov chain Monte Carlo estimation of exponential random graph models". In: *Journal of Social Structure* 3.2 (2002), pp. 1–40.

[49] Olaf Sporns. *Networks of the Brain*. MIT Press, 2011.

[50] Cornelis J. Stam. "Functional connectivity patterns of human magnetoencephalographic recordings: a "small-world" network?" In: *Neuroscience Letters* 355 (2004), pp. 25–28. DOI: 10.1016/j.neulet.2003.10.063.

[51] Laura E Suarez et al. "A connectomics-based taxonomy of mammals". In: *eLife* 11 (2022), e78635.

[52] Laura E. Suarez et al. *MaMI dataset (Version 1) [Data set]*. 2022. DOI: 10.5281/zenodo.6376544. URL: https://doi.org/10.5281/zenodo.6376544.

[53] Franco Vazza and Alberto Feletti. "The quantitative comparison between the neuronal network and the cosmic web". In: *Frontiers in Physics* 8 (2020), p. 525731.

[54] Nguyen Xuan Vinh, Julien Epps, and James Bailey. "Information theoretic measures for clusterings comparison: is a correction for chance necessary?" In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 1073–1080.

[55] Stanley Wasserman and Philippa Pattison. "Logit models and logistic regressions for social networks: I. An introduction to Markov graphs and p". In: *Psychometrika* 61.3 (1996), pp. 401–425.

[56] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world' networks". In: *Nature* 393.6684 (1998), pp. 440–442.