Analysis of information dynamics in protein interaction networks across the tree of life

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Pavitra Batra

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

April, 2023

Supervisor: Dr. Manlio De Domenico

Certificate

This is to certify that this dissertation entitled Analysis of information dynamics in protein interaction networks across the tree of lifetowards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Pavitra Batra at Indian Institute of Science Education and Research under the supervision of Dr. Manlio De Domenico, Associate Professor of Applied Physics, Department of Physics and Astronomy of University of Padua , during the academic year 2022-2023.

Dr. Manlio De Domenico

Committee:

Dr. Manlio De Domenico

Dr. Deepak Dhar

This thesis is dedicated to my family

Declaration

I hereby declare that the matter embodied in the report entitled Analysis of information dynamics in protein interaction networks across the tree of life are the results of the work carried out by me at the Department of Physics and Astronomy of University of Padua, Indian Institute of Science Education and Research, Pune, under the supervision of Dr. Manlio De Domenico and the same has not been submitted elsewhere for any other degree.

Pavitra Batra

Acknowledgments

I would like to thank my supervisor Prof. Manlio De Domenico for his guidance and encouragement. I am grateful for his constant support throughout my thesis. I am deeply indebted to Prof. Deepak Dhar for guiding me during my journey at IISER, particularly for having the patience to indulge in all kinds of discussions, however random the topic be. His unique insights on various aspects have greatly shaped my approach to research and life in general. I would also like to thank Prof. M.S Madhusudhan for being there for advice and help beyond academics. Finally, I would like to thank my family and friends here at IISER who were always there for me.

Abstract

Protein interaction networks are ubiquitous in the functioning of organisms. Inspired by the work of Leskovec et al. on changes in the resilience of such networks, we observe how quantitative characteristics of protein interaction networks change over the evolutionary scale. We find that the spectrum of the Laplacian of the network has features that are similar for similar species, and this correlation can be used to guess the biological genera of species, only knowing its protein network. We then generate a clustering of species using a metric for comparison between different networks. We are currently working on observing how different such a generated tree is from the tree of life generated using sequence data. The thesis follows the following plan:

Chapter 1 We start by introducing protein interaction networks and discussing why their study is important. We then give the motivation for our study, describing the work of Leskovec et al. on the resilience of the network and how it has inspired our work. Finally, we give a brief description of the aim of our study.

Chapter 2 covers all the necessary background theories used. We broadly discuss three broad aspects: the study of networks, using statistics for working with datasets, and the workings of Phylogenetic Trees. In this chapter, we develop our problem in detail and discuss the ideas we used to study the problem at hand.

In *Chapter 3* we discuss some of the existing results which we reproduce in particular the calculation of spectral entropy of some synthetic networks and real divergence between real data. We move to get the spectral entropy for our data and then discuss our exploration of the spectrum of the Laplacian, and finally, come up with a hierarchical clustering to quantify if our method can be extended to generate trees similar to the existing phylogenetic tree.

With *Chapter 4*, as a conclusion, we summarize all the methods and results. We then discuss the limitations of our study and its potential.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

Evolution has shaped a massive diversity of life on the planet. We often want to study these organisms in the light of evolution which will be the underlying idea of the thesis as well. The organisms exhibit several phenotype and biological functions which are the results of the interaction between several molecular components and their environment. All of these interactions between molecules such as DNA, RNA, and proteins can be represented as networks. Such biological networks have gained a lot of attention in recent years[17]. There are several classified networks like the protein-protein interaction network and transcription factor-target regulation networks. As the interactions evolve, so do the networks. We will be focusing on the physical protein-protein interaction map. A lot of work has been done on DNA sequences to understand gene functions and their evolution but there is still a lot not known about how the changes lead to the rewiring of the protein interaction network. Multiple studies have been able to map the interaction between proteins using high through-put experiments such as affinity purification [9, 4] and yeast two-hybrid systems[16, 15].

Amongst many of the works on protein interaction networks, we were motivated by the results of Leskovec et al.[19] on evolution of resilience in protein interaction networks across the tree of life. The authors observe that interactomes became more resilient to network failures along an evolutionary scale(Fig 1.1). We briefly describe their work here. Resilience in these interactomes is characterized by measuring how much the interactome fragments on random removal of some fraction f of the total nodes. Say if a graph G_f has k isolated components on removal f fraction of the total nodes from the original network. The connectivity of this graph is quantified using

normalized Shannon diversity.

$$H_{msh}(G_f) = -\frac{1}{\log N}\sum_{i=1}^{k} p_i \log p_i$$

where N is the number of proteins in the network and $p_i = |C_i|/N$ is the proportion of proteins in the i^{th} component. p_i can also be interpreted as the probability of seeing a protein from component C_i when sampling proteins from the fragmented interactome. $\frac{1}{N}$ is the normalization factor compensating different network sizes. It is evident from the definition that the higher the fragmentation higher the entropy. Thus, the resilience of the network is defined by measuring the entropy over a range of fragmentation rates

$$Resilience(G) = 1 - H_{msh}(G_f)df$$

The following result is obtained when we calculate the resilience for the available species

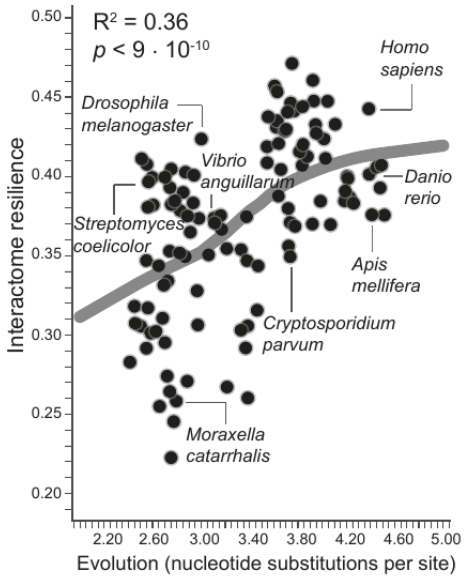


Figure 1.1: Interactome resilience for 171 species with at least 1,000 publications in the NCBI Pub Med (LOWESS fit; R2 = 0.36). Figure taken from [19])

This very interesting result raised several questions on whether we observe similar changes in other properties associated with these networks. We decided to observe if there are evident patterns in how interactions occur given a network and how the interactions are affected due to the changes in topological connections of species over the evolutionary time scale. We will start by replicating

some results to validate the code. We quantify the networks using specific quantitative measures such as a distribution, using these numbers to compare networks. We also test our methods of comparing networks across evolutionary scale by using them against synthetic network models. Finally, We use our metric to form a hierarchical clustering which is like a phylogenetic tree. We compare this hierarchy against the clustering generated using the protein sequences. All of this is done over the same interaction data set as used by Leskovec et al.

Chapter 2

Methods and Background

For all the topics described here, a lot more details can be given we only describe the parts which are directly relevant to the results we obtain, putting the relatively longer proofs or indirect results in the appendix.

2.1 Networks

A network, for us, is a collection of vertices and edges which can be used to represent interaction in some system. For example (see Fig 2.1), if we draw edges between a set of people that know each other with the people as nodes, we get a social network. These mathematical objects allow us to model the system in a simple manner and gain insights into it.

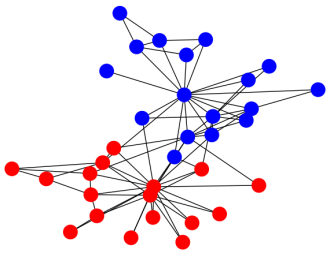


Figure 2.1: Friendship network between members of two different karate clubs. Data from Zachary [18]

We will be using the term network and graph interchangeably. A graph will be represented as a pair $G = (V, E)$

2.2 Laplacian of graph

The Laplacian matrix is another form of representation of a graph beside the adjacency matrix. For an undirected, unweighted network is an $n \times n$ symmetric matrix $\mathbf{L}$ with the components given as

$$L_{ij} = \begin{cases} k_i & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ if an edge b/w } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}$$

where k_i is the degree of node i. We can write this further as

$$L_{ij} = k_i \delta_{ij} - A_{ij}$$
$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

where $\mathbf{D}$ is a diagonal matrix with node degrees as its diagonal entries.

A primary part of the project is dealing with the Laplacian matrix as it turns up in diffusion over networks which we will see later. Thus, we want to know some of its relevant properties.

2.2.1 Properties of Laplacian

1. Every row of Laplacian sums to zero

This can be shown very simply starting with $L_{ij} = k_i \delta_{ij} - A_{ij}$

$$\sum_j L_{ij} = \sum_j (k_i \delta_{ij} - A_{ij}) = k_i - k_i = 0$$

2. Eigenvalues of Laplacian are non-negative real

Laplacian is a real symmetric matrix and therefore has real eigenvalues. Further, taking any eigenvalue λ and its corresponding normalized eigenvector $\mathbf{v}$. Thus, we have $\mathbf{L}\mathbf{v} = \lambda \mathbf{v}$,

$\mathbf{v}^T\mathbf{L}\mathbf{v} = \lambda$ and we proceed as follows

$$\begin{aligned}\sum_{ij} A_{ij}(v_i - v_j)^2 &= \sum_{ij} A_{ij}(v_i^2 - 2v_iv_j + v_j^2) \\ &= \sum_i k_i v_i^2 - 2\sum_{ij} A_{ij}v_iv_j + \sum_j k_j v_j^2 \\ &= 2\sum_{ij}(k_i\delta_{ij} - A_{ij})v_iv_j = 2\sum_{ij} L_{ij}v_iv_j = 2\mathbf{v}^T\mathbf{L}\mathbf{v}\end{aligned}$$

Since the LHS is always non-negative all eigenvalues are non-negative.

3. At least one of its eigenvalues is zero

From the first property, every row sums to zero making the vector with all entries as 1 an eigenvector with eigenvalue zero

2.3 Network Models

2.3.1 Erdos-Renyi Model

Given a graph with n nodes and probability p with which we may place an edge between any two nodes. The ensemble of random graphs we can generate are called Erdos-Renyi graphs. Each graph G in this ensemble appears with probability

$$P(G) = p^m(1-p)^{\binom{n}{2}-m}$$

where m is the number of edges in the network. ER model is one of the simplest forms of random graphs while they fail to capture a lot of features about the working system. They are of importance as a building block and as synthetic network to test our methods over.

2.3.2 Configuration Model

Configuration networks are generalized random graphs, which are good reference models for studying real-world networks. This is primarily because they can have heterogeneous degrees of the nodes. An ensemble of configuration networks corresponding to a given degree distribution

is generated either by rewiring or randomly selecting two stubs(incomplete edges) at a time. Working with configuration networks can also give us insights into how resilient a network is against rewiring. We briefly describe the method of how we generate the configuration networks here

Algorithm

We use a standard library from graph-tool [12] to obtain the ensemble of configuration models. The algorithm iterates through all the edges in the given network and tries to swap its target or source with the target or source of another edge. The choice of sample edges is implemented using the Metropolis-Hastings acceptance/rejection algorithm. The vertex degree distribution converges for a sufficiently large number of iterations. A description of how the Metropolis-Hastings algorithm works is given in the appendix.1.2.

2.3.3 Barabasi-Albert Model

Many networks found in everyday life, like the world wide web, citation networks, etc have degree distributions that follow power laws tail end. Even protein-protein interaction networks have been shown to exhibit similar scale-free behavior scale [7]. People have devised ways of generating such networks with mechanistic incite to generate such scale-free networks. The method used to generate them is commonly called preferential attachment. We will use this model as the synthetic network due its relevant properties.

2.4 Dynamics on Networks

We can give a general form of an equation for a dynamical system with a single variable on a network as

$$\frac{dx_i}{dt} = f_i(x_i) + \sum_j A_{ij} g_{ji}(x_i, x_j)$$

here the first term, in some sense, represents the intrinsic dynamics, and the second term on the right gives the effect from the bonds. That said, we are primarily interested in diffusive dynamics on networks. Starting with Fick's laws of diffusion, we have the diffusion flux is proportional to

the negative of the concentration gradient

$$\frac{\partial c}{\partial t} = -D\Delta c$$

where Δ is the Laplace operator. We can extend this to the networks in the following way. Assume some concentration c_i on i^{th} site and concentration c_j at the j^{th} site which is a neighbor to the i^{th} site. We also take all the constants to be 1 then the equation will be given as

$$\begin{aligned}\frac{dc_i}{dt} &= -\sum_j A_{ij}(c_i - c_j) \\ &= \sum_j (\delta_{ij}k_i - A_{ij})c_j \\ \frac{dc}{dt} &= -\mathbf{L}c\end{aligned}$$

Domenico et al. extend the idea of such dynamics and give a field theory for the same. We describe the framework here.

2.4.1 Framework

Start with a vector space such that its dimension equal to the number of nodes N. Nodes of the network are identified to the basis vectors of this vector space (only the basis vectors are identified, i.e. sum of two nodes does not have to be a node). Basis vectors, i.e. the nodes, are represented as $\langle i|$ where i ranges from 1 to N. The connections are then encoded by a time varying operator $\hat{W}(t)$ which represents the weighted and directed adjacency matrix.

Domenico et al.[6] assume a field $\langle\phi|$ on this vector space spanned by the nodes $\{\langle i|\}$ as basis. This field is called the information field as depending on the system of study it can represent bits of information like small packets of energy or concentration of signaling molecules. Depending on the choice of a F, the system can take up several forms of dynamics. We take a general linearized form of this field on the network as

$$\partial_t \langle\phi(t)| = \langle\phi(t)|F(t,\hat{W}(t))$$

where $\hat{W}(t)$ is the adjacency matrix encoding the edges of the graph. The solution for the above

can be given using a propagator for the dynamics as

$$\langle \phi(t)| = \langle \phi(0)|\hat{S}(t)$$

We assume the initial conditions that the information seed is on any of the i^{th} nodes with uniform probability $p_i = \frac{1}{N}$,i.e we have $\langle \phi(0)| = \sum_{i=1}^{N} p_i \phi_0 \langle i|$. In such a situation the expected flow of information from the i^{th} node is given as $\frac{\phi_0}{N}\langle i|\hat{S}(t)$the j^{th} node to i^{th} node or the contribution of the field value at t^{th} node due to j^{th} node is

$$\frac{\phi_0}{N}\langle i|\hat{S}(t)|j\rangle$$

For systems where the propagator is diagonalizable, we can write it as

$$\hat{S}(t) = \sum_{l=1}^{n} s_l(t)\hat{\sigma}^l(t)$$

where s_l is the eigenvalue associated with the l^{th} eigenvector and σ_l is the outer product of l^{th} right and left eigenvectors. On substituting this form to the expected flow between one pair of nodes we see there are N operators dictating the flow and thus we call these operators $\{\hat{\sigma}^l(t)\}$ as information streams. Figure 2.2 represents these streams on nodes.

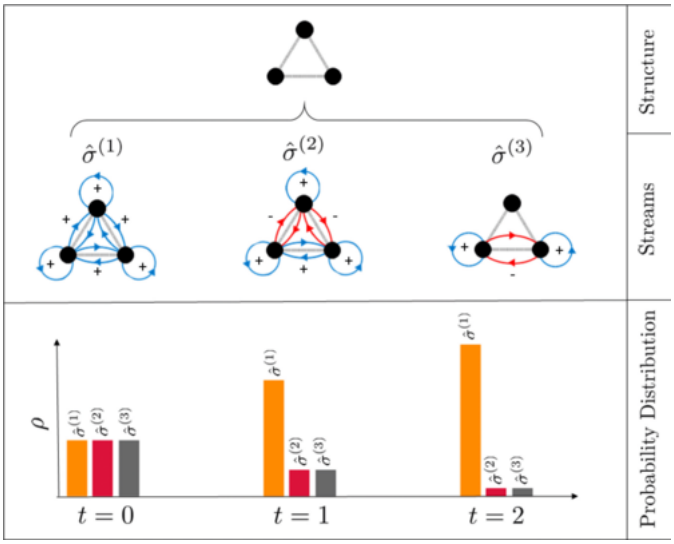


Figure 2.2: Ensemble of information streams for random walk dynamics. A simple system of three fully connected constituents. The figure illustrates the information streams and their corresponding activation probabilities changing over time. Figure taken from [6]

A self-loop traps some part of the field on the node itself that we get by the term $\langle i|\hat{\sigma}^l|i\rangle$. With the stream size $\{\frac{\phi_0}{N}s_l(t)\}$ being equal to the expected trapped field on top of all nodes (Appendix .1.1) overall expected trapped field is the summation of all stream sizes $\sum_{l=1}^{N}\frac{\phi_0}{N}s_l(t) = \frac{\phi_0}{N}Z(t)$. To study the dynamics of the complete field we can restrict ourselves to the study of trapped field as it is directly related to the stream size. We can then describe any dynamics by taking a relevant superposition of the streams.

$$\begin{aligned}\hat{\rho}(t) &= \sum_{l=1}^{N}\rho_l(t)\hat{\sigma}^{(l)}(t)\\ &= \frac{\hat{S}(t)}{Tr(\hat{S}(t))}\end{aligned}$$

The coefficient ρ_l gives the weight associate with each stream, $\rho_l = \frac{\frac{\phi_0}{N}s_l(t)}{\frac{\phi_0}{N}Z(t)}$.

Further, these weights can take up a probabilistic interpretation of discretizing the field. Assuming the field to be discretized into infinitesimal small quanta with a large number of them contributing to the stream size. The number of participating packets is given as $n(t)h = \frac{\phi_0}{N}Z(t)$ and for the l^{th} stream $n^l(t)h = \frac{\phi_0}{N}s_l(t)$. Thus, $\rho_l = \frac{n^l(t)}{n(t)}$ gives the probability that one quanta participates in the activation of l^{th} stream. Thus we have an additional probabilistic interpretation to the eigenvalues of the propagator S

We model protein interaction networks by assuming that proteins interact in accordance with their concentration gradients and follow diffusive dynamics. This can be translated back to the framework described in the previous section

$$\partial_t\langle\phi(t)| = -\langle\phi(t)|\mathbf{L}$$

and the propagator then is $\hat{S} = e^{-\tau\mathbf{L}}$, similarly, we get $\hat{\rho}$, which is given the name density matrix [5]

$$\hat{\rho} = \frac{e^{-\tau L}}{Tr(e^{-\tau L})}$$

As we can see, we want to work with the density matrix as it governs the flow over the network. For this, we want to study this matrix thus, we will study its characteristic values, i.e., eigenvalues extensively.

2.5 Statistics

As clear from the discussion above, we will be working extensively with probability distributions and would want ways to study them and often compare one with another.

2.5.1 Comparing Distributions

The distributions that we will be working with come from real network data. The distributions cannot be assumed to be coming from the same underlying populations. We can neither make assumptions about the populations being governed by some parametric form such as the normal distribution. Thus, we will rely on non parametric hypotheses tests.

Rank Sum test

We start with two distributions of sizes m and n, respectively. Rank the two $n+m$ values from both samples in ascending order. Label one of the samples as first does not matter which. The test statistic(TS) for the test is defined as the sum of the rank of the data in the first sample. Assuming H_0 to be the hypothesis that two of the populations are identical and say that the test statistic TS has value t. To reject the null hypothesis with significance value α for the two sided test we have

$$P(TS \leq t) \leq \frac{\alpha}{2}$$

or

$$P(TS \geq t) \geq \frac{\alpha}{2}$$

where both the probabilities are calculated under the assumption that H_0 is true. We reject the null hypothesis if the p value given by the data set is less than or equal to α where the p value is given as

$$p \text{ value} = 2Min(P(TS \leq t), P(TS \geq t))$$

In order to know the probabilities, we need to know the distribution to which the TS belongs to when H_0 is true. When the null hypothesis is true we know that the $n+m$ values and any random

selection of n values as a sample will belong to the same distribution using this one can show that when H_0 is true.

$$E[TS] = \frac{n(n+m+1)}{2}$$
$$Var(TS) = \frac{nm(n+m+1)}{12}$$

Additionally, when both n and m are both big enough the test statistic has approximately a normal distribution. Thus, allowing us to calculate the properties. For any ties in the ranking, the average of the ranks is allotted to all the values with the same rank. This test is also called the Mann-Whitney test.[14]

Kolmogorov Smirnov Test

Again we start with two populations and one measurable characteristic, say same biological species and its height. We want to know if the two sample sets are statistically identical. KS- test proceeds the following way. The samples of observations are $a_1, \ldots, a_n$ and $b_1, \ldots, b_m$. For every t denote by $A(t)$ the fraction k/n of subscripts i for which $a_i \leq t$. The function defined in this way is the empirical distribution of the a's. Similarly we define the distribution for the distribution b's. The KS statistic for the comparison is given as

$$TS = \sup_t |A(t) - B(t)|$$

where $A(t)$ and $B(t)$ are the empirical distribution functions of the samples. The probability of statistic compared against the Kolmogorov distribution with appropriate significance value. An intuitive way of understanding the test when the number of samples are same is by associating with the samples a path of length $2r$ leading from origin to the point $(2r, 0)$. If the samples are indistinguishable, the sampling makes all the possible paths equally probable. [3]

Kruskal-Wallis Test

While we described the Whitney-Mann test, which works for two distributions, we often work with more than two distributions. Say we have n populations each with distribution P_i of the property of interest of the i^{th} population. The null hypothesis will be given as $H_0 : P_0 = P_1 = \ldots = P_n$. The

alternative hypothesis being that not all distributions are equal. The test statistic for the test is given as

$$TS = \sum_{i=1}^{n} \frac{R_i^2}{k_i}$$

where R_i is the sum of the ranks of sample corresponding to the $i^{th} population$ and k_i is the sample size of the i^{th} population. To figure out the significance value we use the approximation that for sufficiently large k_i the distribution

$$\frac{12}{K(K+1)} TS - 3(K+1)$$

follows a chi-squared distribution with $n-1$ degrees of freedom. Rejecting H_0 if the above distribution is greater than equal to the corresponding value of the chi-square distribution.

2.5.2 Entropy

Entropy of a random variable gives a measure of the uncertainty in the possible outcomes. For a given random variable X entropy is defined as a function of its probability distribution. One classical measure of entropy of a random variable is the Shannon entropy which is give as

$$H(X) = -\sum_{x} p_x \log p_x$$

where by convention $0 \log 0 \equiv 0$. A good intuition about this definition is that it can be used to quantify the resources needed to store information.[11]

Von Neuman entropy is the extension of the classical Shannon entropy and is used to define the entropy of quantum states. For a quantum state ρ the Von Neuman entropy of the quantum state

$$S(\rho) = -tr(\rho \log_2 \rho)$$

and if λ_x are eigenvalues of ρ then the above can be reduced to .2.1

$$S(\rho) = -\sum_{x} \lambda_x \log_2 \lambda_x$$

Spectral Entropy

Domenico et al. define a matrix corresponding for a given network G as

$$\hat{\rho} = \frac{e^{-\tau \mathbf{L}}}{Z}$$

where $\mathbf{L}$ is the Laplacian of the network, τ is constant parameter interpreted as taking role of time and $Z = Tr(e^{-\tau \mathbf{L}})$. The matrix is called the density matrix and is defined by the motivation that it needs to follow the properties of density matrix from quantum mechanics such as being positive semi-definite. The Von Neuman Entropy for this matrix is called the spectral entropy for network

$$S(t) = -Tr(\hat{\rho}(t) \ln \hat{\rho}(t))$$

this can be further simplified using $\lambda_i(\rho) = Z^{-1} e^{-\beta \lambda_i(\mathbf{L})}$ as

$$\begin{aligned} S(\rho) &= -\sum_{i=1}^{N} \lambda_i(\rho) \log_2 \lambda_i(\rho) \\ S(G) &= \frac{1}{Z \ln 2} \sum_{i=1}^{N} e^{-\beta \lambda_i(\mathbf{L})} [\ln Z + \beta \lambda_i(L)] \\ &= log_2 Z + \frac{\beta}{\ln 2} Tr[\mathbf{L}\rho] \end{aligned}$$

where G is the corresponding network.[1] The spectral entropy defined this way overlaps with the Von Neuman entropy of the information streams described above under continuous diffusion dynamics. Through either of the descriptions the entropy is of the measure of diffusion across the network. Higher the entropy more de-localized the diffusion is, and lower the entropy higher the localization of information is. Further, one can use the Laplacian directly as well for the density matrix, but entropy from such density matrix violates subadditivity.

Relative Entropy

Relative entropy is a measure of closeness between any two given probability distributions. Say we have two distributions $p(x)$ and $q(x)$ defined over the same domain space with same number of

indices, the relative entropy is defined as

$$H(p(x) \parallel q(x)) \equiv \sum_{x \in \mathbf{X}} p(x) \log \left(\frac{p(x)}{q(x)} \right)$$

Relative entropy is also called Kullback-Leibler Divergence, and is often used to compare sample distribution against a model probability distribution. While relative entropy is a good measure of closeness, it has some limitations in the way we can use it to compare distributions. Starting with the first issue is that it is not a distance metric and which can be used for clustering. Additionally, it is not symmetric. Thus, we instead use one of its variant.

Jensen Shannon Divergence

Jensen Shannon Divergence is a variation of Kullback-Leibler divergence and is given as

$$\text{JSD}(P \parallel Q) = \frac{1}{2} D(P \parallel M) + \frac{1}{2} D(Q \parallel M)$$

where $M = \frac{1}{2}(P+Q)$. The square root of the Jensen-Shannon divergence is the distance metric [6].

2.6 Protein Networks

Before discussing any details of our analysis of the protein interaction networks. It is a good idea to get a sense of what are protein interaction networks. While proteins do interact chemically sometimes the primary mode of interaction between proteins is physical contact. Proteins also interact with several other bio molecules but when we say protein interaction networks we restrict to only one protein interacting with other proteins. All such interactions form the protein interaction network. Thus, a node on the network is protein and if they interact we put an unweighted edge between them.

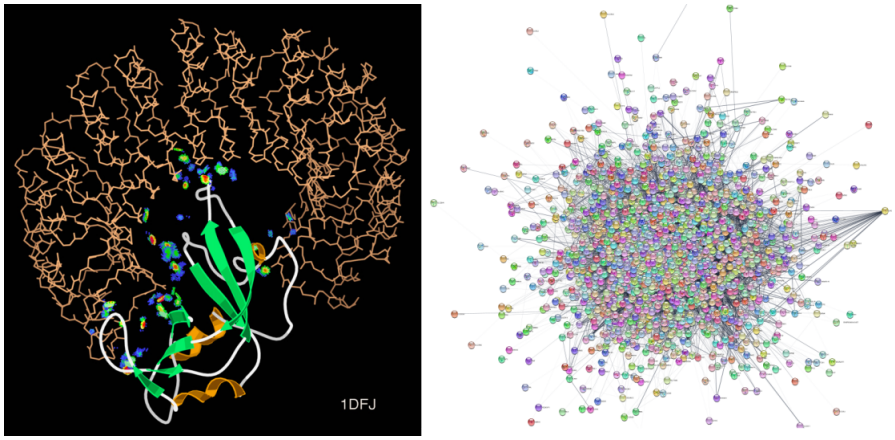


Figure 2.3: a. The horseshoe shaped ribonuclease inhibitor (shown as wire frame) forms a protein–protein interaction with the ribonuclease protein. The contacts between the two proteins are shown as colored patches.(left; taken from Wikipedia) b. Protein interaction network of Homo Sapiens generated using our data(right)

2.6.1 Data Set

We use the data set compiled by Leskovec et al. which we filter for some parts. We will briefly describe the data set being used here for a detailed discussion please refer to the supplementary information provided by Leskovec et al. The data set is publicly available at SNAP library.

In terms of numbers we start with 1840 species (1539 bacteria, 111 archaea, 190 eukarya) involving $14,50,633 proteins$ (1.4 mil) and $87,62,166$ protein interactions (8.7 mil). Only experimentally supported and human expert curated physical(direct) protein-protein interactions are considered to build the interactomes. Physical or direct interactions include regulatory interactions, binary interactions, signaling interactions, kinase-substrate pairs, metabolic enzyme-coupled interactions and protein complexes. Indirect functional interactions are dropped. We use these interactions to create undirected unweighted networks. The data has been obtained from the STRING database. Two salient features of the data sets are that one the datasets is quality controlled and that it is species-specific as no computationally predicted datasets are included or by orthology.

There are several biases associated with the protein data said, with only a limited number of interactions known for the widely studied species. Leskovec et al. have broadly classified biases into types *a*. Inter species data bias and *b*. intra species data bias.

Inter-species Bias Data associated with model organisms is much more widely studied and preserved which leads to considerable variation in the quality and amount of data kept for species.

Intra-species Bias In a species often highly expressive gene-coded proteins in certain cell lines get focused upon more so than others leading to further variability.

To address some of the biases in the protein interaction data set associated with the inter-species bias, the original authors use data such that the publication count greater for interactome of species is greater than 1000 in order to prevent biasing from tail end less studied. We work with 404 species allowing species with publication count greater than 100 and the largest connected component of each of the interactomes to include some additional tail end organism and see how the results change. In order to make data usable for us we convert the edge list in the form of protein names to numbers and prepare a dictionary of the same for reference.

Tree of life as described by Hug et al. is used to characterize evolution in each of the species. Given a species s, its evolution t_s is calculated as the total branch length (i.e., nucleotide substitutions per site) from the root of the tree to the leaf representing species s.[19]

2.7 Phylogenetic Trees

Phylogenetic trees are hierarchical clustering that represent the evolutionary history of organisms. Such phylogenetic trees can either be rooted or unrooted. While it is common to work with unrooted trees when it comes to biological species as we seek to get an idea of the relationship between different species and are not sure about the existence of a single starting point.

Maximum Likelihood tree

Maximum likelihood is a popular method of phylogeny inference. One particular implementation of maximal likelihood is followed by RAxML-NG[8], which we use in order to generate our tree. Maximum likelihood relies on Bayesian inference. Some basic features of the trees that we generate are 1. they are unrooted 2. they are binary. To take an example, say we have 4 taxa the number

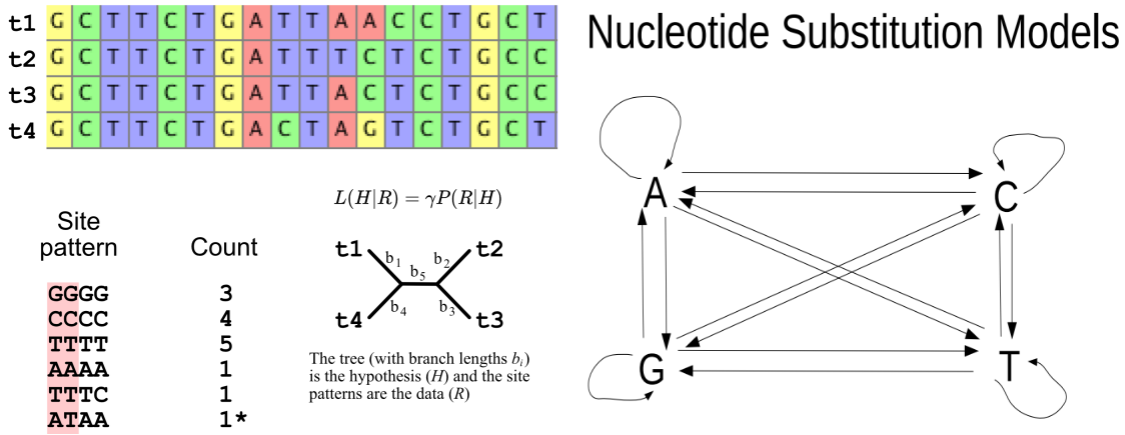


Figure 2.4: (Left) Illustration of the phylogenetic tree generation through likelihood methods[Taken from Wikipedia].(Right) Illustration of nucleotide substitution model [Taken from presentation slides of the RAxML authors]

of possible unrooted trees for this situation is 3. The method generates all possible trees and assigns each of them a score. Before moving to give the scoring criteria, it is important to know that the number of possible trees grows exponentially (there are almost 10^{67} trees when we work with 2000 taxa) thus it is not sensible to look through all the possible trees. Instead, the program follows a greedy logic and selects the local maxima rather than the global maxima. The scoring criteria are given by using the following idea. The gene/alignment sequence associated with the species is taken. This sequence is assumed to evolve following time-reversible Markov chains i.e. each site in a sequence evolves and mutates to some other nucleotide. Then per-site likelihood is calculated for each sequence. We start with a parsimonious tree and then we apply lazy sub tree arrangements such as grafting and pruning.

2.7.1 Our Clustering

We use a sequential, agglomeration, hierarchic, non-overlapping method(SAHN) for obtaining a bifurcating tree of the organisms. This is implemented using the Scipy linkage library. We describe the details of the algorithm here.

We input the pairwise distance between N points, which here are the JSD between any two modified eigenvalue spectra of the density matrices of species. The output that we get is a step wise dendrogram. The step wise dendrogram data structure is defined in the following way:

Definition 2.7.1. *A step wise dendrogram is a list of $K-1$ triples $(m_l, n_l, \delta_l)(l = 0, \ldots, K-2)$ such that $\delta_l \in [0, \infty)$ and K is the cardinality of of a given set S_0. Further, $m_l, n_l \in S_l$, where S_{l+1} has been recursively defined as $(S_l \backslash \{m_l, n_l\}) \cup k_l$ and $k_l \notin S \backslash \{m_l, n_l\}$ is a label for a new node.*

The set S_0 is the initial set of data points. At every step, we add a new node labeled k_l, which is the hypothetical parent of m_l and n_l. The set S is then updated by adding this new label, and its children (m_l, n_l) are removed. m_l and n_l are selected .The result is a node containing all initial nodes. The distance for this particular new node can be updated through different choices. We similarly make choice of new pairs to cluster based on their distance. [10] The pseudo code for the primitive algorithm is given here. The exact algorithm differs in the implementation and has some additional properties, in particular, being faster and maintaining a list of neighbors for each node. The details of the original algorithm can be read in detail from the work of Mullner.[10]

Algorithm 1 Working definition of a hierarchical clustering(Taken from [10])

procedure PRIMITIVE _CLUSTERING(S, d) ▷ S : node labels, d : pairwise dissimilarities
  $K \leftarrow |S|$ ▷ Number of input nodes
  $L \leftarrow []$ ▷ Output list
  $size[x] \leftarrow 1$ for all $x \in S$
  for $l \leftarrow 0, \ldots \{K-2\}$ **do**
    $(m, n) \leftarrow argmin_{(SxS) \backslash \Delta} d$
    Append $(m, n, d[m, n])$ to L
    $S \leftarrow S \backslash \{m, n\}$
    Create a new node label $k \notin S$
    Update d with the information

$$d[k, x] = d[x, k] = Formula(d[m, x], d[n, x], d[m, n], size[m], size[n], size[x])$$

  for all $x \in S$.
    $size[k] \leftarrow size[m] + size[n]$
    $S \leftarrow S$
  end for
  return L ▷ the step wise dendrogram as $((K-1)) \times 3$-matrix)
end procedure

(As usual, Δ denotes the diagonal in the Cartesian product)

The clustering algorithm we use is an improvement over this primitive clustering algorithm The primitive clustering is considered to be the benchmark for the algorithm which we use. A schematic of the primitive algorithm can be given as follows

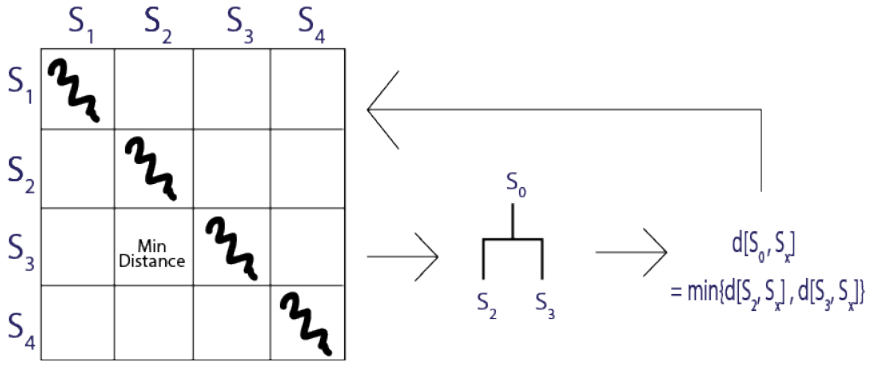


Figure 2.5: Schematic of the primitive algorithm. We club the two species with minimum distance between their distributions. Introduce the clubbing as a new leaf and update the distance between this new leaf and all other existing species. Repeat this process. Obtain a binary tree

2.7.2 Data Set

We pick the organisms common in work of Leskovec et al. and Hug et al. There are situations where there are ”two” different organisms belonging to the same species but having different strains. For our purposes, we identify them as the same common organism and use them for comparison. After filtering the organisms, we find that there are around 180 species common to the two data sets. We want to generate a tree for these 200 species using the method used by Hug et al. They generate the tree in the following way. Alignments generated from the SSU rRNA genes of the species are obtained. SSU rRNA genes longer than 600 bp are aligned, the alignment is stripped of columns containing more than 95%. They then put this data to generate maximum likelihood tree using RAxML with relevant parameters. The tree generated in this way is mostly congruent with those generated using ribosomal protein sequences. We use the same method to generate a tree for our work. We compare this tree against the tree generated using the clustering algorithm described above.

Chapter 3

Results and Discussion

We reproduce some of the existing results on synthetic networks before applying our techniques to real-world data. This helps us verify the working of the code. All the results against which the working of code is compared are from the work of Domenico et. al [1]. Almost all the codes by us are publicly available here.

We implement spectral entropy calculations for the Erdos-Renyi network for various link probabilities, p. For each value of p we average over 10 samples of 30 possible graphs of size 50. The original paper does the sampling for higher numbers of network size 200 but since the aim of the exercise was to only verify the working, in the interest of time we reduced the numbers.

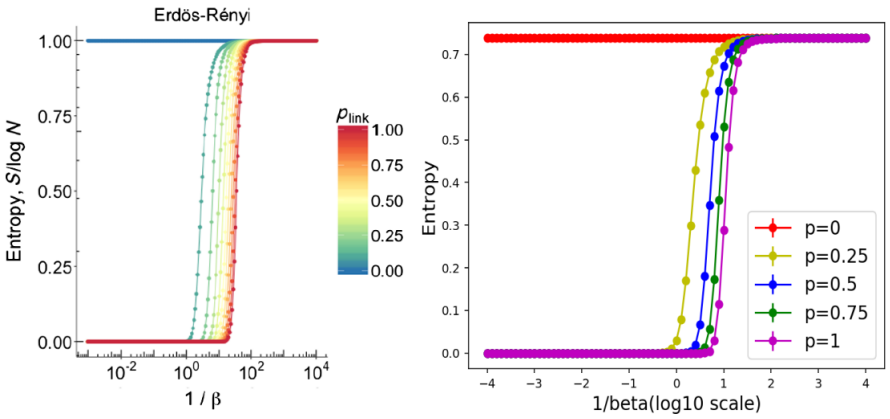


Figure 3.1: Spectral entropy as a function of $\frac{1}{\beta}$ for Erdös-Rényi networks from the paper (left) and my code (right)

Next, we use the multi-layer networks based on structure and function built from the human microbiome. This consists of 18 layers each one corresponding to a body site. These layers have been partitioned into community types, by using Dirichlet multinomial mixture models, that may be associated with complex diseases [2]. We compare the Jensen-Shannon distance between each pair of layers for different for $beta = 0.1$ as calculated by Domenico et al. for the same against ours.

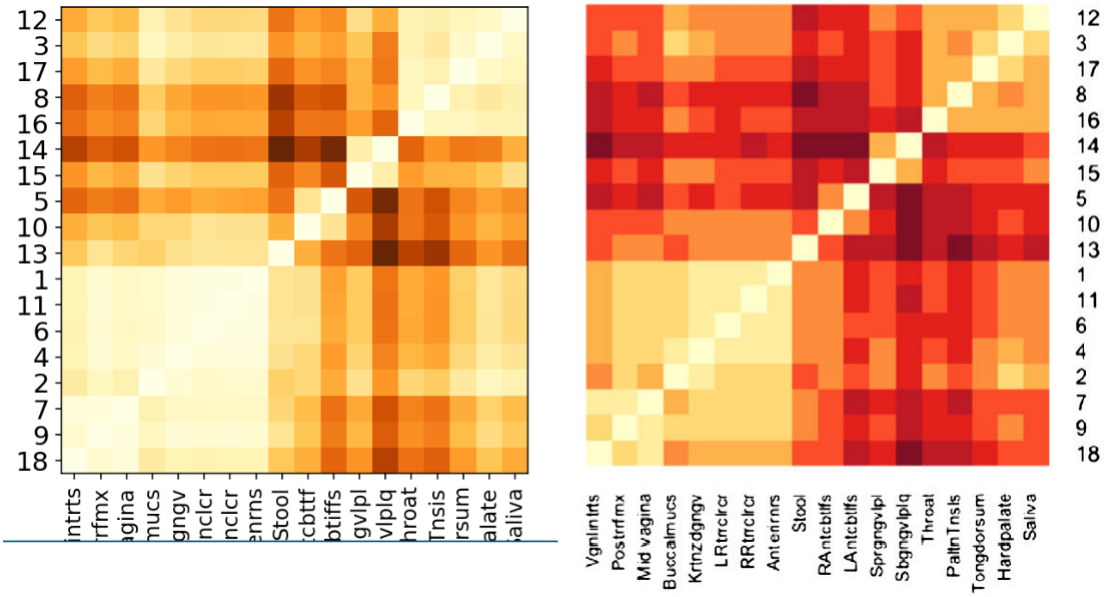


Figure 3.2: Hierarchical clustering of human microbiome sites. The Jensen-Shannon distance matrices with $\beta = 0.1$ from the paper (right) and my code (left)

In both the cases we are able to reproduce the results as can be seen by comparing the trend in the first case and the community clusters in the second. We do similar verification for calculating spectral gap and KL divergences.

3.1 Protein Interaction Networks

The modules of code which could be verified against available data were verified. We now move to working with the results of the protein interaction network data. We started with calculating the normalized spectral entropy of networks of the species plotting it against the evolutionary scale. We use the evolutionary scale as defined in the above chapters and obtained the following results.

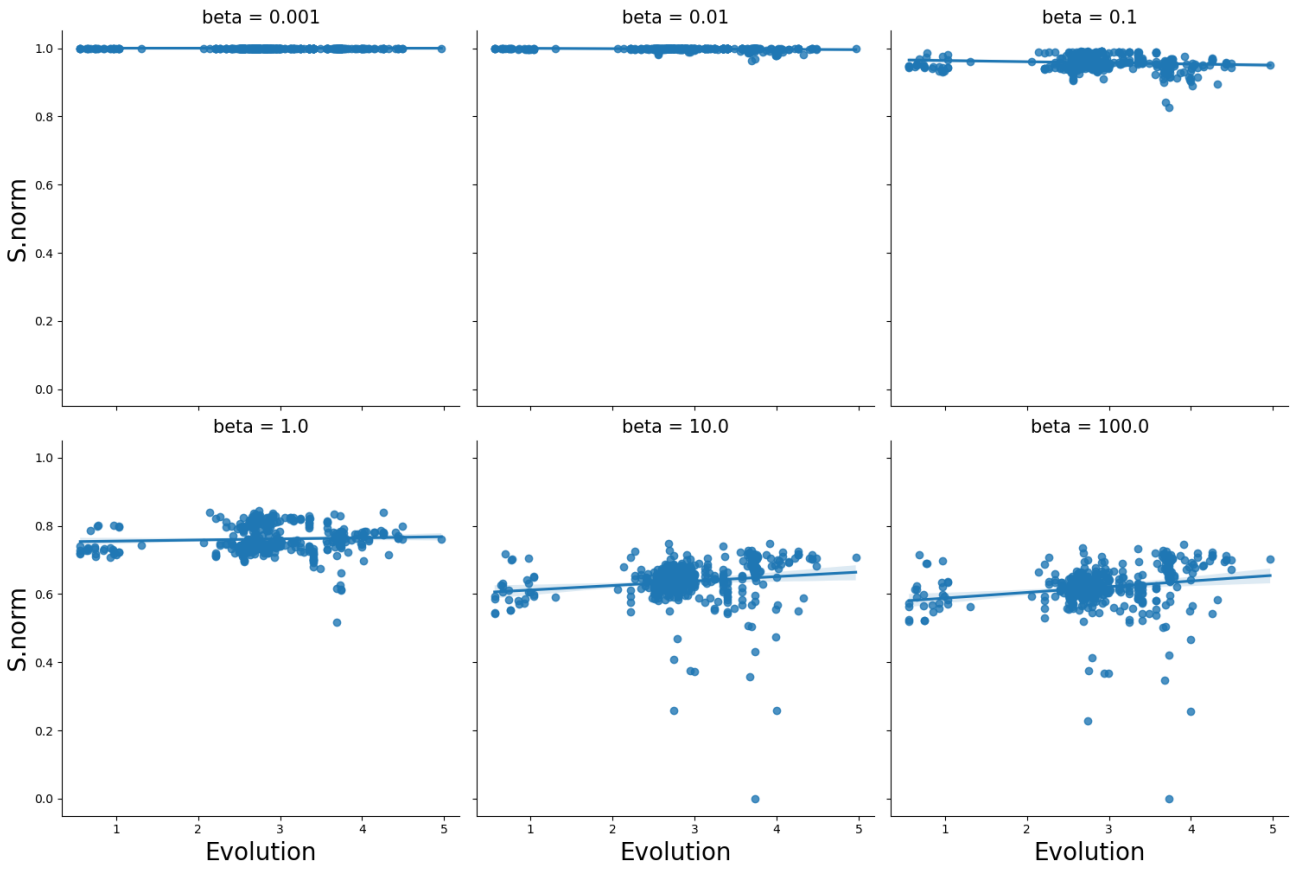
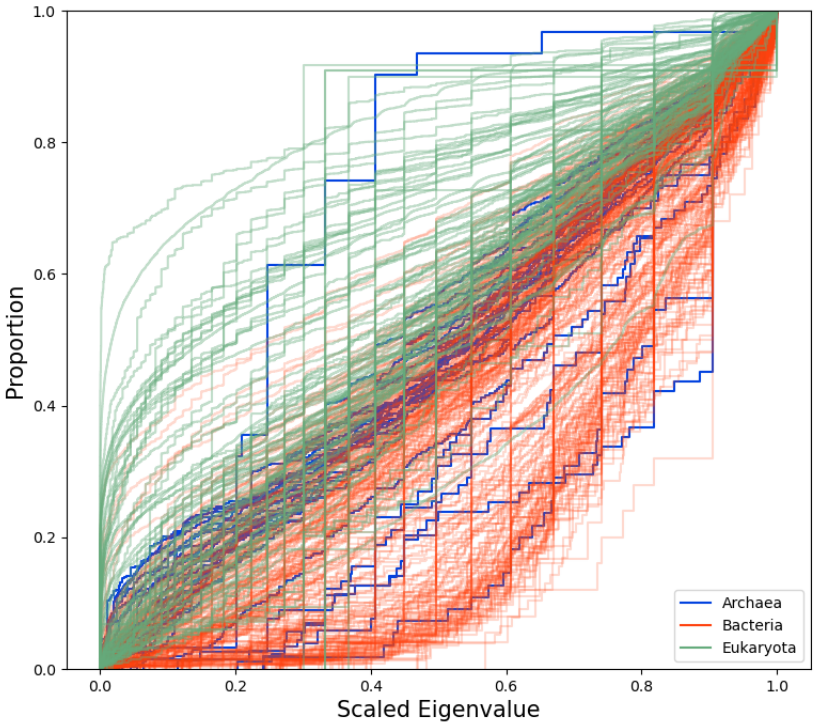


Figure 3.3: Variation in entropy against evolution

β here is the time scaling constant τ. As we see, the spectral entropy decreases with the time constant indicating that the overall information flow across networks gets more localized. No difference is seen in flow across the evolutionary scale, indicating that the overall flow does not seem to have changed.

With not much to observe in the spectral entropy, we start with more basic characteristics of the network. We start with plotting the eigenspectra of the species. Instead of directly plotting the eigenvalues we plot the cumulative frequency distribution of eigenvalues of density matrix normalized against their maxima.



The idea of observing on an individual species scale was put on hold and instead a domain-based comparison was adopted as it made more sense to start at a slightly more coarse-grained level. On plotting the above graph, we observed a very interesting transition in domains of the scaled eigenvalues around the line $y = x$. In order to quantify these observations we started with distance measures and Kolmogorov type of test.

3.1.1 Network Comparison

We want to observe if we can compare two different kinds of networks. When networks are of the same size, it is a fairly straightforward process we generate a distribution of the same number of values in the same space. Thus we can directly apply the Jensen Shannon Divergence over the two distributions. We start with comparing the eigenvalue distributions against two types of distributions, one a uniform probability distribution and second the distribution of configuration networks. In the case of configuration networks corresponding to each species we generate 20

configuration networks and the average of divergence against the 20 of these networks. Following are the results that we obtain

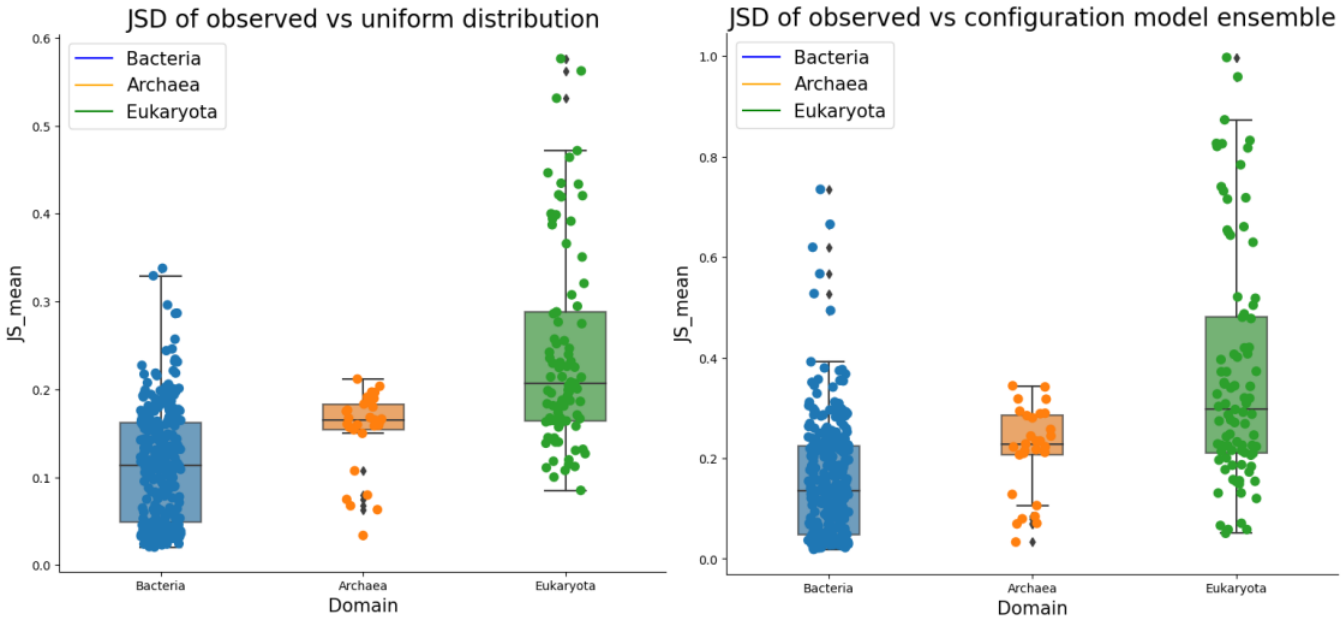


Figure 3.4: Distance of each species against different distributions. Against uniform distribution on left and against its configuration networks on right.

Visually the mean divergence of bacteria is fairly distinct from that of archaea and Eukaryota. To quantify this visual difference we perform the Whitney-Mann test in pairs and find that, all three populations are distinct. Additionally, we also perform the Kruskal-Wallis test which also distinguishes the three populations stating that at least one of the observations is different. According to the currently accepted tree of life, Archaea and Eukaryota share more recent ancestry as compared to bacteria. This point is also evident from the closeness of Eukaryota to the archaea. The above results are common for both comparisons against the uniform distribution and the configuration models. The second set of information to draw from the above analysis is that the distribution of eigenvalues are not flat. An important point to note is that all comparisons have been done against $\beta = 0.1$ for the density matrix. This is primarily due to the observations from a separate set of runs where most meaningful results are found for $\beta \leq 1$ and $\beta \approx 0.1$

Before moving forward, we want to verify if such a comparison method works by using the method to compare synthetic networks. We perform two experiments for this. In first, we take 50 samples each of the Barabasi-Albert Model and Stochastic Block Model. We then generate 50 configuration model instances for each network. On calculating Jensen Shannon divergence

between the original network and its corresponding 50 configuration models and taking the mean value of the divergence we observe the following

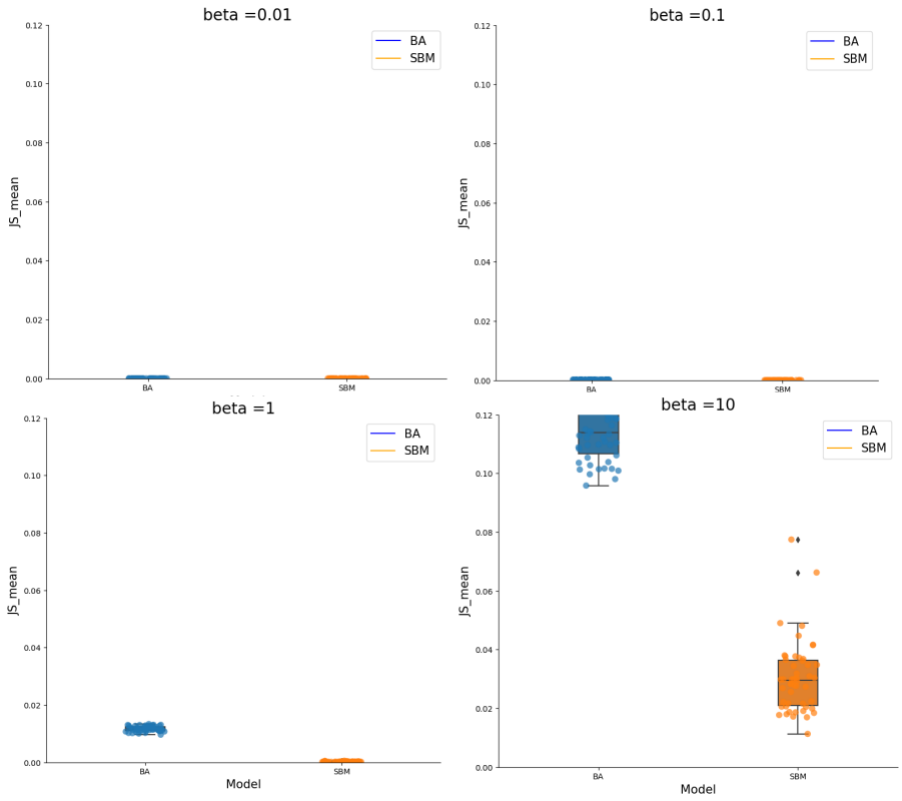


Figure 3.5: The mean js-div between original networks and their configuration networks. BA on left SBM on right for several beta

We provide the visual representation of eigenvalue distribution in the appendix.4 As can be seen from either the visual representation of the distribution or the above graph the synthetic networks generated from such rewiring resemble closely their original ones in regards to diffusive properties. This is expected as the degrees remain the same as node identities don't matter much even when connections change over them.

Moving to the second run. We generate 50 iterations of each BA and SBM 1000 node for both types of models and then generate 50 more BA and SBM with same parameters. Calculating JS divergence of the newly generated model against the ones generated in first step and taking mean value we obtain the following

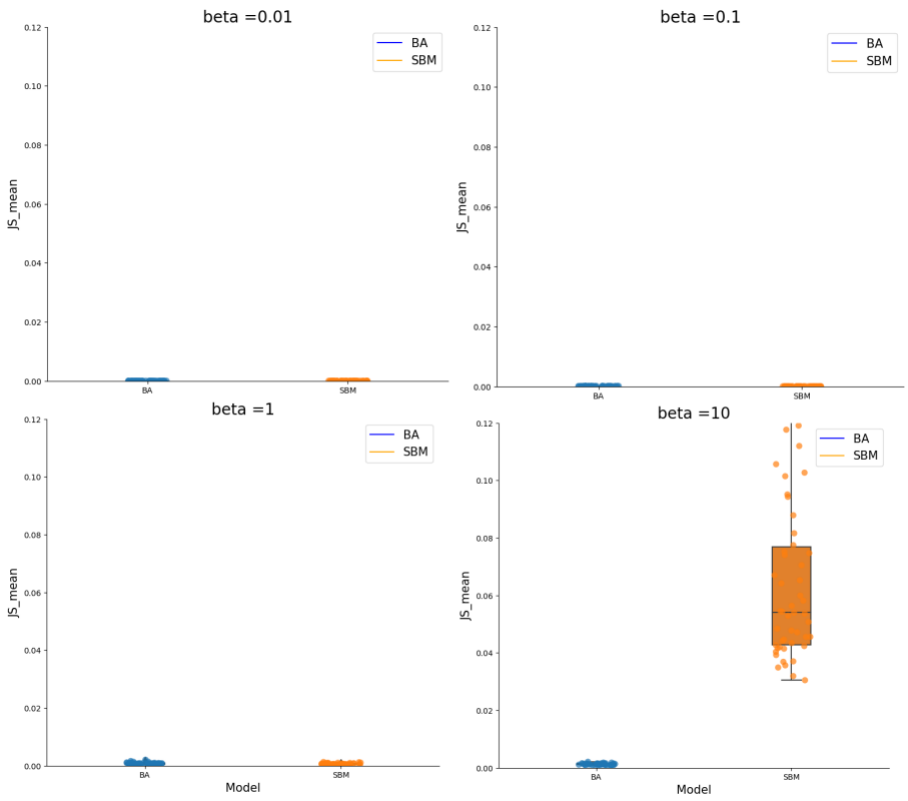


Figure 3.6: The mean js-div between original networks and the newly generated networks. BA on left SBM on right for several beta

The result for this run is again very similar to the previous run and bear the same explanation. Finally, for beta > 1 the results seem to get getting skewed. This can be possibly due to lower-valued eigenvalues starting to take values in very small order(< -10).

Until now, all the analysis was restricted to using networks of the same sizes, but this does not allow us to do a direct comparison of the species. With all species having different network sizes, we want to be able to compare them directly. The definition of the density matrices allows us to do this. With the eigenvalues of density matrix restricted to the same domain i.e $\in [0, 1]$ we are allowed to do a comparison of them. We do this comparison in the following way we bin the distributions by keeping the number of bins around the average size of all networks that is 1000 (the exact average is 788.3875). Once the interval is binned we see their occurrence frequency in a particular bin. By this method all distributions are in same space and with same number of indices. This allows us to use the Jensen Shannon divergence on these distributions now. Applying this method for comparing all the species with each other we observe the following

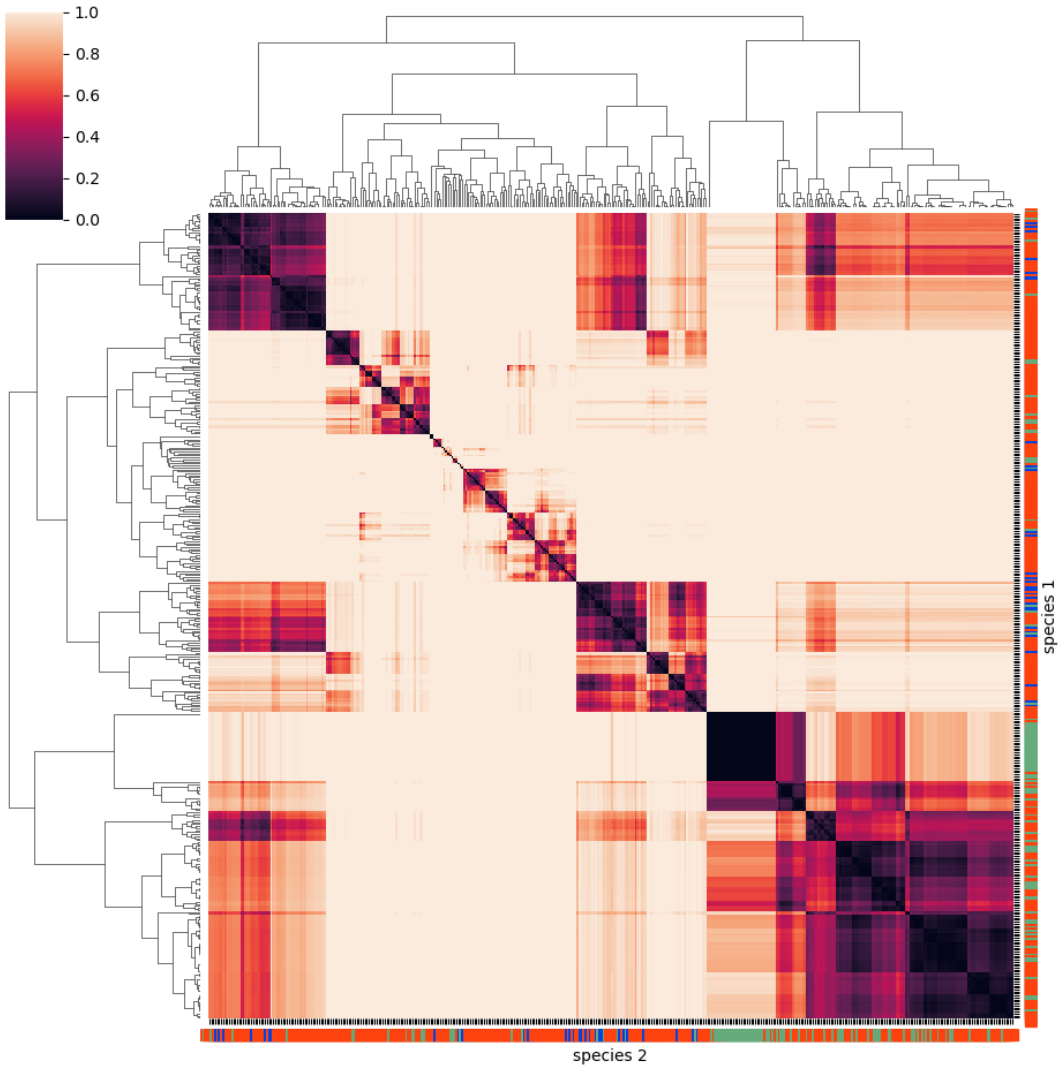


Figure 3.7: Distance of each species against different distributions. Against uniform distribution on the left and against its configuration networks on right.

A significant amount of clustering is seen with species from the same domain being much more closer than the ones from different domain. This process additionally has allowed us to give a distance metric between any two networks. This matrix allows us to cluster the species using a SAHN clustering method and create a realization of phylogenetic tree.

We can summarize the above method briefly using the schematic(Fig 3.8), we start with a network of N nodes use the function of Laplacian. Take the eigenvalues of the new matrix. Do this for networks of all species. All the eigenvalues lie in the same domain and we bin this domain to

get distributions of the same size and then take the distance between the distributions.

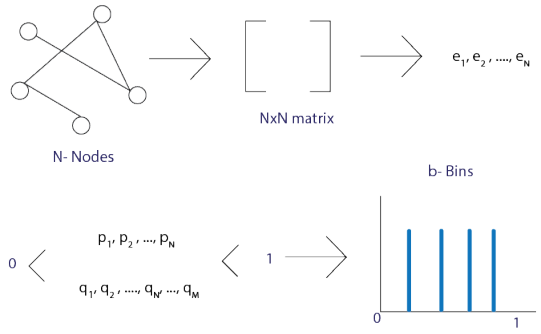


Figure 3.8: Schematic of comparing network of different sizes.

Now that we have a distance metric between all the distributions, we generate a SAHN clustering. We also take the sequence data for organisms from the same species to generate the maximum likelihood tree. The figure below gives a visual comparison between the SAHN algorithm and the tree generated using maximum likelihood. Visually the clustering obtained from SAHN does not seem to be very similar to the tree generated using maximum likelihood. We are currently working on using a quantitative metric to compare the two clustering to allow us to validate our results.

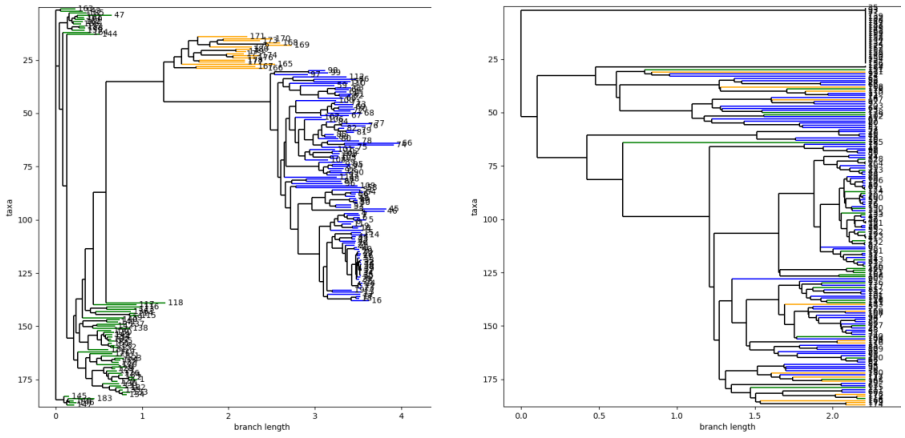


Figure 3.9: Tree generated using RAxML(left) and SAHN(right)

A different version of the above trees with collapsed branches is shared below for better visualization.

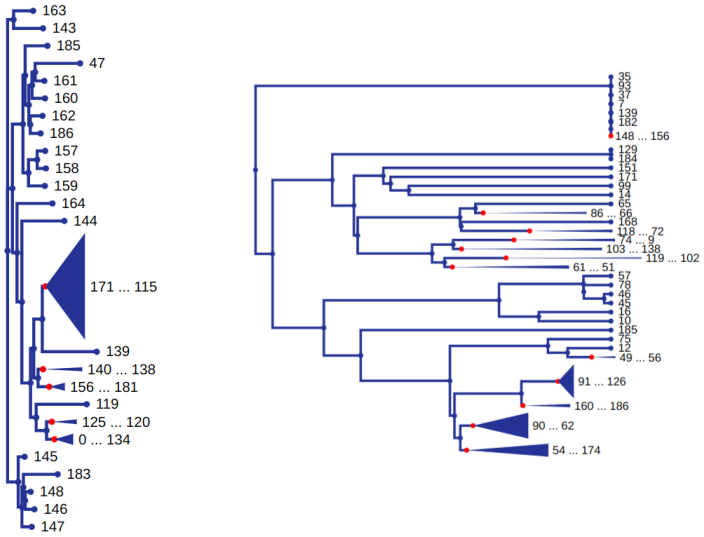


Figure 3.10: Same comparison but for better visualization with help of phlo.io [13]

Chapter 4

Summary and Outlook

The study of biological networks has, in recent times, garnered a lot of attention. This involves studying the network architecture, associated evolutionary mechanisms, and other features of such networks. Of particular interest to us are the protein interaction networks. Inspired by Leskovec et al.'s work on changes in the resilience of such networks across the evolutionary scale, we decided to observe how protein interactions modeled as diffusion change over both the evolutionary scale and the scaling constant β. We find that using the spectrum of the density matrices of the network of a species, we can, after all, classify the species into their evolutionary domains. Then we moved to study how this distribution looks like and giving a distance measure between all the distributions allowing us to recreate a "tree of life" of our own. We are currently working on observing how different such a generated tree is from what the original tree of life is.

While a lot of analysis is still ongoing, there are several ways of making the already obtained results more concrete and a lot of new possible directions to work on. The protein interaction networks that we use are not complete mappings, with a lot of protein mappings missing. Additionally, the protein network structures described themselves are subject to errors in experimental measures, and the edges often have some associated confidence values. Incorporating such issues using some stochastic or ensemble approach could be possible but was out of the scope of the project. We can also try to work with a much more realistic interaction modeling. This modeling does not necessarily detail but is rather specific, perhaps using something like weighted networks or boolean/directional networks. In conclusion, there are a lot of possible avenues to explore and study, with some very interesting results already observed on our system.

Appendices

.1 Networks

.1.1 Dynamics on Networks

Framework

We can get the expected amount of trapped field by l^{th} stream in the following way.

$$\begin{aligned}\frac{\phi_0}{N}\sum_{i=1}^{N}\langle i|\hat{S(t)}|i\rangle &= \frac{\phi_0}{N}Tr(S(t)) \\ &= \frac{\phi_0}{N}\sum_{i=1}^{N} \\ &= \frac{\phi_0}{N}s_l(t)\end{aligned}$$

Here, the trace of each of the stream operators is one since they are the outer product of the left and right eigenvectors of propagators.

.1.2 Network Model

Metropolis-Hastings Algorithm

The metropolis-Hastings algorithm is a Markov Chain Monte Carlo method for obtaining a random sample from a probability distribution. We have a probability distribution say $p(x)$ from which we wish to sample. We create a Markov chain of samples. Starting with $f(x)$ which is a simpler form of $p(x)$ where $p(x) = \frac{f(x)}{N}$

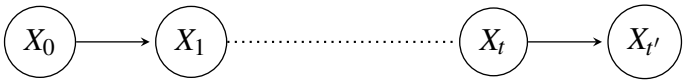


Figure 1: Markov Chain of samples

The initial set of samples would not be very useful, but the samples that we will obtain later in time should resemble the desired distribution. To do this we start with a candidate easier candidate

distribution i.e the normal distribution

$$g(X_{t+1}|X_t) = N(X_t, \sigma^2)$$

This function is dependent on the immediately previous sample and does not have to be normal. Selecting a symmetric normal distribution is called the Metropolis algorithm and using an asymmetric distribution as the Metropolis-Hastings algorithm both follow similar idea but differ in the following sense. We will be describing the Metropolis algorithm. We take the previous sample X_t, take a normal distribution centered around this and select the next sample from this normal distribution. The next step once we have a new candidate for the sample is whether we accept or reject this candidate. We accept this candidate with $P(X_t \to X_{t+1})$. The acceptance probability is obtained using the idea of detailed balance. We have the following condition

$$f(m)g(n|m)P(m \to n) = f(n)g(m \to n)P(n \to m)$$

From the above condition we can get

$$P(m \to n) = Max\Big(\frac{f(b)}{f(a)}\Big)$$

We thus have the Markov chain constructed and can now get the desired distribution.

.2 Statistics

.2.1 Entropy

Von Neumann Entropy

Von Neumann Entropy for a density matrix is given as

$$S(\rho) = -Tr(\rho \log_2 \rho)$$

which in turn is

$$S(\rho) = -\sum_i^N \lambda_i log_2 \lambda_i$$

by convention $0\log 0 = 0$

Proof:

$$Tr(A) = \sum_j \langle j|A|j\rangle$$

The following are the important ideas that we use

1. Trace of a matrix is same independent of the basis we can choose any basis

2. Density Matrix ρ is hermitian therefore orthonormal eigenvector always exist

3. $\rho = \sum_i^N \lambda_i |i\rangle\langle i|$ is the spectral decomposition of the density matrix with eigenvalues λ_i's and eigenvectors $|i\rangle$

4. Choose the basis to be orthonormal eigenvectors.

$$\begin{aligned}
Tr(\rho \log_2 \rho) &= \sum_i \langle i|\rho \log_2 \rho|i\rangle \\
&= \sum_i \langle i|\rho \left((\rho - I) - \frac{(\rho - I)^2}{I} + \frac{(\rho - I)^3}{I} \dots\right)|i\rangle \\
&\text{since } \rho|i\rangle = \lambda_i|i\rangle \text{ we have} \\
&= \sum_i \langle i|\rho \left((\lambda_i - I) - \frac{(\lambda_i - I)^2}{I} + \frac{(\lambda_i - I)^3}{I} \dots\right)|i\rangle \\
&= \sum_i \langle i|\rho \log_2 \lambda_i|i\rangle \\
&= \sum_i \log_2 \lambda_i \langle i|\rho|i\rangle \\
&= \sum_i \lambda_i \log_2 \lambda_i \langle i|i\rangle \\
&= \sum_i \lambda_i \log_2 \lambda_i
\end{aligned}$$

check condition on 1. ρ for the Taylor expansion 2. Obtained λ_i

.3 Phylogenetic Trees

.3.1 Tree Formats

Newick tree format or New Hampshire tree format is a popular format for writing trees in a readable format. The specification of the file format is given as follows;

Each terminal leaf of a tree represents a species. All strings end with a semi-colon. The leaves in the outermost hierarchy are the oldest/closest to the root node. Any objects inside the brackets and separated by a coma are at the same level of hierarchy. Thus, a leaf described by the string "$(((A,B),(C,D)),(E,F,G));$" will look something like this

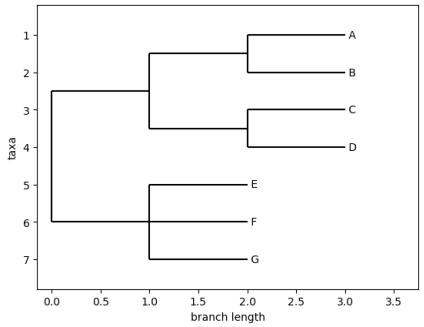


Figure 2: Example of a Newick tree

We can incorporate branch lengths by adding the length against the species separated by a colon("$(A:0.1,(B:0.2,C:0.3));$"). We describe the specifications in detail as some of the code involved converting the Newick Tree format into our linkage matrix for us to compare two clustering.

.4 Results

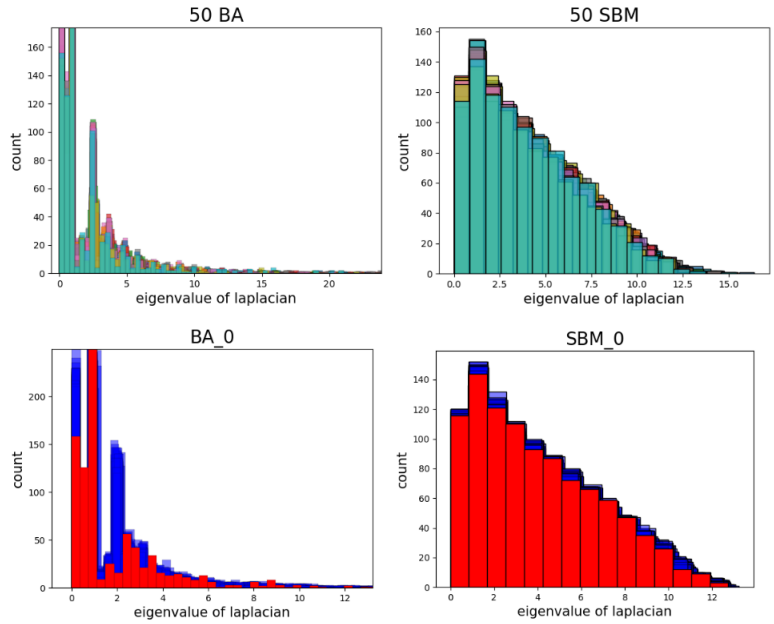


Figure 3: Distribution of eigenvalues from the first run described. (Top Left) Eigenvalues for the 50 BA models (Bottom Left) Eigenvalue distribution for one particular model and its configuration models. Similarly, for the SBM model on the right.

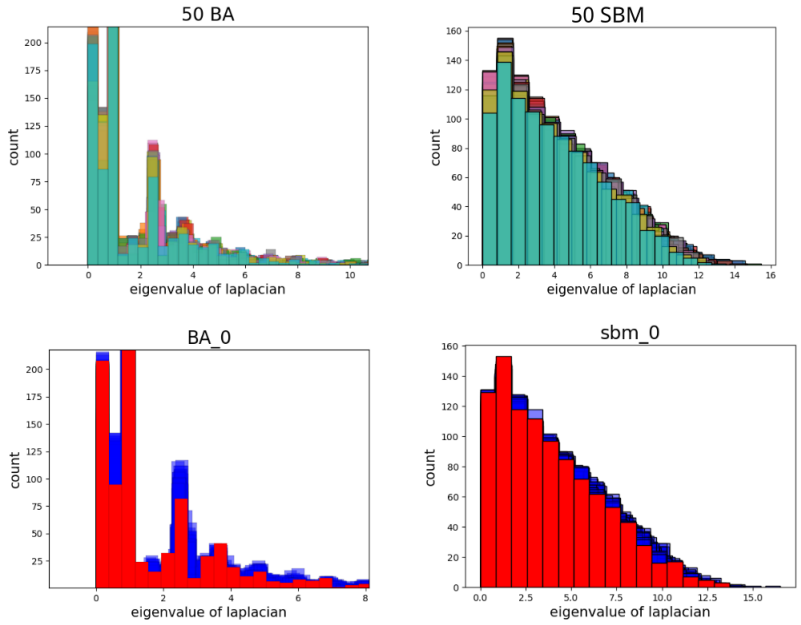


Figure 4: Distribution of eigenvalues from the second run described. (Top Left) Eigenvalues for the 50 SBM models (Bottom Left) Eigenvalue distribution for one particular model and configuration (Top right) (Bottom right)

Bibliography

[1] Manlio De Domenico and Jacob Biamonte. "Spectral Entropies as Information-Theoretic Tools for Complex Network Comparison". In: *Phys. Rev. X* 6 (4 Dec. 2016), p. 041062. DOI: 10.1103/PhysRevX.6.041062. URL: https://link.aps.org/doi/10.1103/PhysRevX.6.041062.

[2] Tao Ding and Patrick D. Schloss. "Dynamics and associations of microbial community types across the human body". In: *Nature* 509.7500 (May 2014), pp. 357–360. ISSN: 1476-4687. DOI: 10.1038/nature13178. URL: https://doi.org/10.1038/nature13178.

[3] William Feller. *An introduction to probability theory and its applications, Vol. 1, 2nd ed.* An introduction to probability theory and its applications, Vol. 1, 2nd ed. Oxford, England: John Wiley, 1957, pp. xv, 461–xv, 461.

[4] Anne-Claude Gavin et al. "Proteome survey reveals modularity of the yeast cell machinery". In: *Nature* 440.7084 (Mar. 2006), pp. 631–636. ISSN: 1476-4687. DOI: 10.1038/nature04532. URL: https://doi.org/10.1038/nature04532.

[5] Arsham Ghavasieh and Manlio De Domenico. *Generalized network density matrices for analysis of multiscale functional diversity*. 2022. arXiv: 2210.16701 [physics.soc-ph].

[6] Arsham Ghavasieh, Carlo Nicolini, and Manlio De Domenico. "Statistical physics of complex information dynamics". In: *Phys. Rev. E* 102 (5 Nov. 2020), p. 052304. DOI: 10.1103/PhysRevE.102.052304. URL: https://link.aps.org/doi/10.1103/PhysRevE.102.052304.

[7] H. Jeong et al. "The large-scale organization of metabolic networks". In: *Nature* 407.6804 (Oct. 2000), pp. 651–654. ISSN: 1476-4687. DOI: 10.1038/35036627. URL: https://doi.org/10.1038/35036627.

[8] Alexey M Kozlov et al. “RAxML-NG: a fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference”. In: *Bioinformatics* 35.21 (May 2019), pp. 4453–4455. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz305. eprint: https://academic.oup.com/bioinformatics/article-pdf/35/21/4453/30330794/btz305_supplementary_data.pdf. URL: https://doi.org/10.1093/bioinformatics/btz305.

[9] Nevan J. Krogan et al. “Global landscape of protein complexes in the yeast Saccharomyces cerevisiae”. In: *Nature* 440.7084 (Mar. 2006), pp. 637–643. ISSN: 1476-4687. DOI: 10.1038/nature04670. URL: https://doi.org/10.1038/nature04670.

[10] Daniel Müllner. *Modern hierarchical, agglomerative clustering algorithms*. 2011. DOI: 10.48550/ARXIV.1109.2378. URL: https://arxiv.org/abs/1109.2378.

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

[12] Tiago P. Peixoto. “The graph-tool python library”. In: *figshare* (2014). DOI: 10.6084/m9.figshare.1164194. URL: http://figshare.com/articles/graph_tool/1164194 (visited on 09/10/2014).

[13] Oscar Robinson, David Dylus, and Christophe Dessimoz. “*Phylo.Io*: Interactive viewing and comparison of large phylogenetic trees on the web”. en. In: *Mol. Biol. Evol.* 33.8 (Aug. 2016), pp. 2163–2166.

[14] Sheldon M. Ross. “CHAPTER 14 - Nonparametric Hypotheses Tests”. In: *Introductory Statistics (Third Edition)*. Ed. by Sheldon M. Ross. Third Edition. Boston: Academic Press, 2010, pp. 647–697. ISBN: 978-0-12-374388-6. DOI: https://doi.org/10.1016/B978-0-12-374388-6.00014-4. URL: https://www.sciencedirect.com/science/article/pii/B9780123743886000144.

[15] Jean-François Rual et al. “Towards a proteome-scale map of the human protein–protein interaction network”. In: *Nature* 437.7062 (Oct. 2005), pp. 1173–1178. ISSN: 1476-4687. DOI: 10.1038/nature04209. URL: https://doi.org/10.1038/nature04209.

[16] Ulrich Stelzl et al. “A Human Protein-Protein Interaction Network: A Resource for Annotating the Proteome”. In: *Cell* 122.6 (2005), pp. 957–968. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2005.08.029. URL: https://www.sciencedirect.com/science/article/pii/S0092867405008664.

[17] Mark GF Sun and Philip M. Kim. “Evolution of biological interaction networks: from models to real data”. In: *Genome Biology* 12.12 (Dec. 2011), p. 235. ISSN: 1474-760X. DOI: 10.1186/gb-2011-12-12-235. URL: https://doi.org/10.1186/gb-2011-12-12-235.

[18] Wayne W. Zachary. “An Information Flow Model for Conflict and Fission in Small Groups”. In: *Journal of Anthropological Research* 33.4 (1977), pp. 452–473. DOI: 10.1086/jar.33.4.3629752. eprint: https://doi.org/10.1086/jar.33.4.3629752. URL: https://doi.org/10.1086/jar.33.4.3629752.

[19] Marinka Zitnik et al. “Evolution of resilience in protein interactomes across the tree of life”. en. In: *Proc Natl Acad Sci U S A* 116.10 (Feb. 2019), pp. 4426–4433.