Graph Neural Network in Computational Biology

M.Sc. MATHEMATICS

2024-2026

Graph Neural Network in Computational Biology

DISSERTATION

Submitted to the University of Kerala in partial fulfillment of the requirements for the award of Master of Science in Mathematics

Programme Code	:	620
Exam Code	:	62023402
Course Code	:	MM 545

CERTIFICATE

This is to certify that the dissertation entitled **"GRAPH NEURAL NETWORK IN COMPUTATIONAL BIOLOGY"** is a record of studies carried out by **RIZVANA R**, M.Sc. Mathematics student of the year **2024-2026** at **T.K.M. College of Arts and Science, Kollam**, under my guidance and submitted to the University of Kerala in partial fulfillment of the Degree of Master of Science in Mathematics.

Kollam

July 2026

Dr. ASWATHY M R

H.O.D. and Assistant Professor

Department of Mathematics

T.K.M. College of Arts and Science

Kollam

HEAD

Department of Mathematics

T.K.M. College of Arts and Science

Kollam - 691 005

Contents

1 INTRODUCTION **1**

2 PRELIMINARY **2**
2.1 Graph Machine Learning 2
2.2 Computational Biology 3
2.3 Biology 3
2.4 DNA in Biology 3
2.5 RNA in Biology 3
2.6 Network in Biology 4
2.7 Protein 4
2.8 Graph 4
2.9 Normalized Laplacian Matrix 5
2.10 Spectrum of the Normalized Laplacian Matrix 6
2.11 Eigenvalue Sets 7
2.12 Biological Networks 7
2.13 Spectrum Plot 7
2.14 Domain 7
2.15 Binary Tree Reconstruction 8

3 The General GNN Framework **9**
3.1 Graph ML for Computational Biology 10
3.2 Biology is Interconnected 11
3.3 Why networks in Biology ? 12

4 The Spectral Analysis on Biology Networks **15**
4.0.1 Normalized Laplacian Matrix 17
4.0.2 Spectrum of Normalized Laplacian Matrix 20
4.1 Eigenvalue sets 21

4.1.1 Eigenvalue Distributions of Two Biological Networks: . 22
4.2 Spectrum Plot 23
4.2.1 Spectrum Plot Graph 24
4.3 Protein and Domain 26
4.4 Domain Co-occurence Networks 27
4.5 How the different Organisms act in Domain Co-occurrence Networks (DCNs) 28
4.6 The Spectrums of Domain Co-occurence Networks 29
4.7 Protein–Protein Interaction Networks of Six Organisms 31
4.7.1 The Spectrums of Protein Interaction Networks 33
4.7.2 Role of Spectral Graphs in Studying Protein Interactions 33
4.8 Binary Tree Reconstruction 34
4.9 Edit Distance 36
4.10 Bipartite Matching Distance 41
4.10.1 Experimental Results 44
4.10.2 Experimental Results 46
4.10.3 Experimental Results 47

Conclusion **50**

Bibliography **51**

Chapter 1

INTRODUCTION

Graph Machine Learning (Graph ML) provides a powerful framework for analyzing biological networks by representing proteins, genes, and other biological entities as nodes and their interactions as edges. Graph Neural Networks (GNNs) and graph-based methods enable the study of complex biological relationships, protein functions, and disease mechanisms. In this project, the Laplacian and normalized Laplacian matrices are used to perform spectral analysis of protein interaction networks. By analyzing eigenvalues, spectrum plots, edit distance, bipartite matching distance, and other graph-based metrics, the structural similarities and differences between biological networks, including those of *Homo sapiens* and *Mus musculus*, are evaluated. These methods provide an efficient approach for comparing, reconstructing, and understanding the topology of biological networks.

Chapter 2

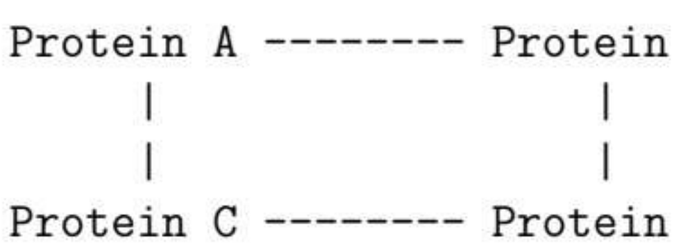
PRELIMINARY

2.1 Graph Machine Learning

Graph Machine Learning (Graph ML) is a machine learning technique that analyzes graph-structured data by representing objects as nodes and their relationships as edges. It is widely used in computational biology to study protein interaction networks, gene regulatory networks, and disease pathways.

Example

Consider the following Protein–Protein Interaction (PPI) network:



In this graph,

- Nodes: Protein A, Protein B, Protein C, Protein D
- Edges: Interactions between proteins

A Graph Machine Learning model can analyze this network to predict protein functions, identify missing protein interactions, and classify proteins based on their connectivity.

2.2 Computational Biology

Computational biology is the branch of science that uses computers, mathematics, and algorithms to analyze and understand biological data.

Example: Analyzing DNA sequences to identify genes associated with a disease.

2.3 Biology

Biology is the branch of science that studies living organisms, their structure, functions, growth, evolution, and interactions with the environment.

Example: Studying the growth and development of plants.

2.4 DNA in Biology

DNA (Deoxyribonucleic Acid) is the genetic material that stores hereditary information in living organisms. It contains the instructions required for the growth, development, reproduction, and functioning of cells.

Actions of DNA

- Stores genetic information.
- Replicates during cell division.
- Directs RNA synthesis through transcription.
- Controls protein synthesis and gene expression.

Example: DNA carries the genes responsible for traits such as eye color and blood type.

2.5 RNA in Biology

RNA (Ribonucleic Acid) is a nucleic acid that carries genetic information from DNA and plays an important role in protein synthesis. It helps convert genetic information into functional proteins.

Actions of RNA

- Messenger RNA (mRNA) carries genetic information from DNA to the ribosome.
- Transfer RNA (tRNA) transports amino acids during protein synthesis.
- Ribosomal RNA (rRNA) forms the structural and functional core of ribosomes.

Example: Messenger RNA (mRNA) carries the genetic code from DNA to ribosomes, where proteins are synthesized.

2.6 Network in Biology

A biological network is a graphical representation of biological components, such as genes, proteins, or metabolites, and the interactions between them. It helps researchers understand how these components work together to perform biological functions and regulate cellular processes.

Example: A protein–protein interaction (PPI) network, where proteins are represented as nodes and their interactions are represented as edges.

2.7 Protein

A protein is a large biological molecule composed of amino acids that performs essential functions in living organisms, including growth, repair, transport, and regulation of cellular processes.

Example: Hemoglobin is a protein that transports oxygen from the lungs to different parts of the body through the blood.

2.8 Graph

A graph is a mathematical structure used to represent relationships between objects. It consists of a set of vertices (nodes) and edges (connections), where the vertices represent objects and the edges represent the relationships between them. Graphs are widely used in computer science, mathematics, and computational biology to model complex networks.

A graph is denoted by

$$G = (V, E), \tag{2.1}$$

where

- V is the set of vertices (nodes),
- E is the set of edges (connections) between the vertices.

Example

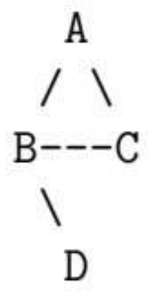
Consider a graph with four vertices:

$$V = \{A, B, C, D\}$$

and the edge set

$$E = \{(A, B), (A, C), (B, C), (B, D)\}.$$

The graph can be represented as



2.9 Normalized Laplacian Matrix

The normalized Laplacian matrix is a normalized form of the graph Laplacian that incorporates the degree of each vertex. It is used to analyze graph connectivity, compare different networks, and perform spectral analysis of biological networks.

$$L_{\text{norm}} = I - D^{-1/2} A D^{-1/2}, \tag{2.2}$$

where

- I is the identity matrix,

- A is the adjacency matrix,
- D is the degree matrix.

Example

For the graph

```
A ----- B
```

the adjacency matrix is

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

and the degree matrix is

$$D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Therefore,

$$L_{\text{norm}} = I - D^{-1/2} A D^{-1/2} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

2.10 Spectrum of the Normalized Laplacian Matrix

The spectrum of the normalized Laplacian matrix is the collection of its eigenvalues, which provides important information about the connectivity and structure of a graph.

Example:

For the normalized Laplacian matrix

$$L_{\text{norm}} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix},$$

the spectrum (eigenvalue set) is

$$\{0,\ 2\}.$$

2.11 Eigenvalue Sets

Eigenvalue sets are the collection of eigenvalues obtained from a matrix, such as the Laplacian or normalized Laplacian matrix. They are used to analyze and compare the structural properties of graphs and biological networks.

Example:

$$\{0,\ 0.52,\ 1.18,\ 2.00\}$$

2.12 Biological Networks

Biological networks are graphical representations of biological entities, such as genes, proteins, or metabolites, and the interactions between them. They help researchers understand the structure and function of complex biological systems.

Example: A protein–protein interaction (PPI) network, where proteins are represented as nodes and their interactions as edges.

2.13 Spectrum Plot

A spectrum plot is a graphical representation of the eigenvalues of a graph matrix, such as the Laplacian or normalized Laplacian matrix. It helps visualize the structural properties and compare different biological networks.

Example: A plot showing the eigenvalues $\{0,\ 0.45,\ 1.20,\ 2.00\}$ of a normalized Laplacian matrix.

2.14 Domain

A domain is a distinct structural and functional region of a protein that performs a specific biological function. A single protein may contain one or more domains.

Example: The SH2 domain binds to phosphorylated proteins and plays an important role in cell signaling.

2.15 Binary Tree Reconstruction

Binary Tree Reconstruction is the process of constructing a binary tree from given data or relationships. In computational biology, it is commonly used to represent evolutionary relationships and hierarchical structures among biological entities.

Example: Constructing a phylogenetic tree to show the evolutionary relationships among different species.

Chapter 3

The General GNN Framework

Graph Neural Networks (GNNs) are a class of deep learning models specifically designed to process graph-structured data. Unlike traditional neural networks, which are suitable for Euclidean data such as images and text, GNNs are capable of learning from the complex relationships among interconnected entities in a graph.

In a graph, entities are represented as

- **nodes (vertices)**, while the relationships or interactions between them are represented as **edges**.

In computational biology, many biological systems are naturally represented as graphs. For example, proteins, genes, cells, and molecules can be modeled as nodes, whereas their interactions, regulatory relationships, or chemical bonds are represented as edges. This graph-based representation enables GNNs to effectively capture both the structural information and biological characteristics of these systems. Consequently, GNNs have become powerful tools for various computational biology applications, including protein–protein interaction prediction, gene regulatory network analysis, disease gene identification, drug discovery, and molecular property prediction.

The general framework of a Graph Neural Network is based on an iterative message-passing mechanism, where each node aggregates information from its neighboring nodes and updates its feature representation. This process allows the network to learn informative embeddings that capture both local neighborhood information and the overall graph structure. The following sections describe the general architecture and working principles of

Graph Neural Networks in detail.

Biological Applications of Graphs

1. Drug Discovery and Development

Applications

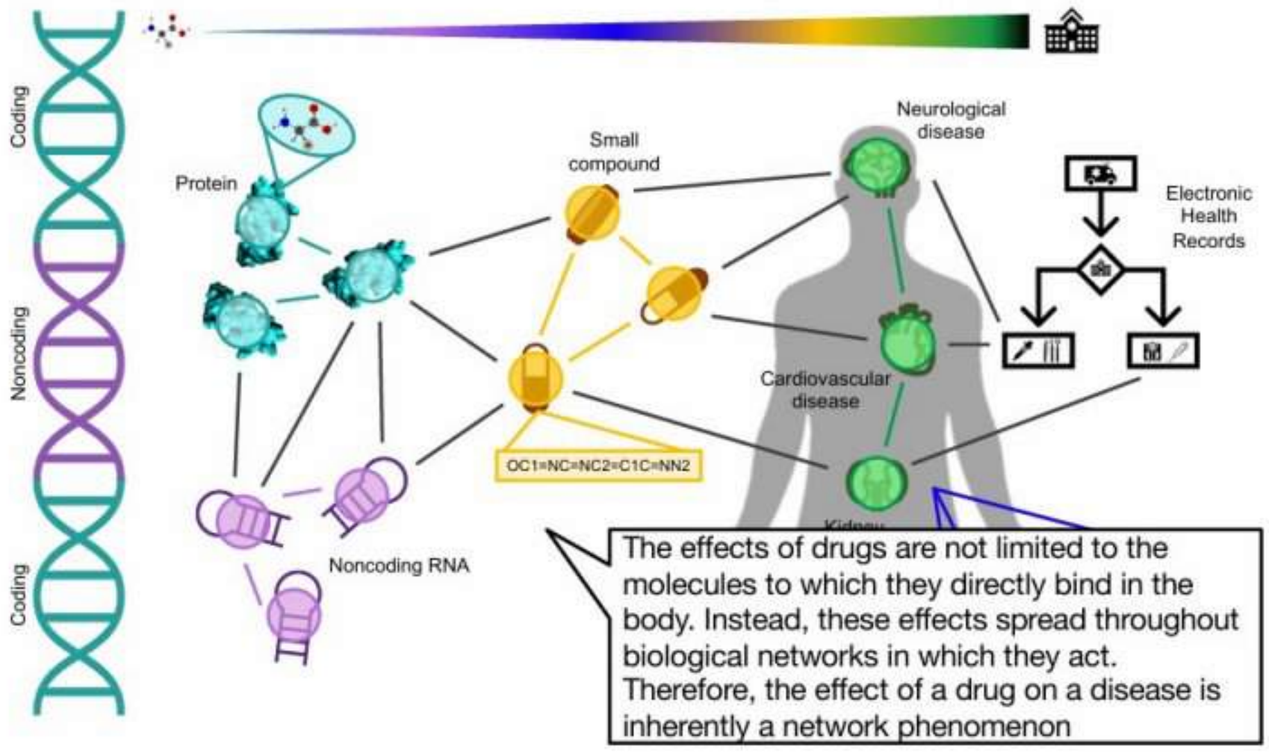
- **Molecular Property Prediction**
  - Predict toxicity, solubility, and binding affinity of small molecules.
- **Drug–Target Interaction**
  - Predict whether a drug molecule binds to a specific protein.
- **Drug Repurposing**
  - Use biomedical knowledge graphs to identify existing drugs that may treat new diseases.
- **Generative Drug Design**
  - Create new molecules by generating their molecular graphs.

3.1 Graph ML for Computational Biology

- There has been a surge of interest in leveraging GNNs for learning meaningful representations of biology.
- GNNs have been used to learn representations that enable critical predictions in downstream applications.

*Machine learning for biomedical networks: Advancements, challenges, and opportunities(2021).

3.2 Biology is Interconnected



The figure emphasizes that the action of a drug is not limited to the molecule it directly binds. Instead, drug effects propagate through complex biological networks involving genes, proteins, RNAs, and organs. Therefore, drug response and disease progression are network phenomena rather than isolated events.

- **DNA**
  - The DNA strand is divided into **coding** and **noncoding** regions.
  - Coding DNA produces proteins, while noncoding DNA gives rise to noncoding RNAs, which also regulate biological processes.
- **Proteins and Noncoding RNA:**
  - Proteins (cyan) interact with one another, forming a protein interaction network.
  - Noncoding RNAs (purple) also interact with proteins and other RNAs, influencing gene expression and cellular functions.
- **Small Compounds/Drugs**
  - Small molecules or drugs (yellow) bind to specific proteins.

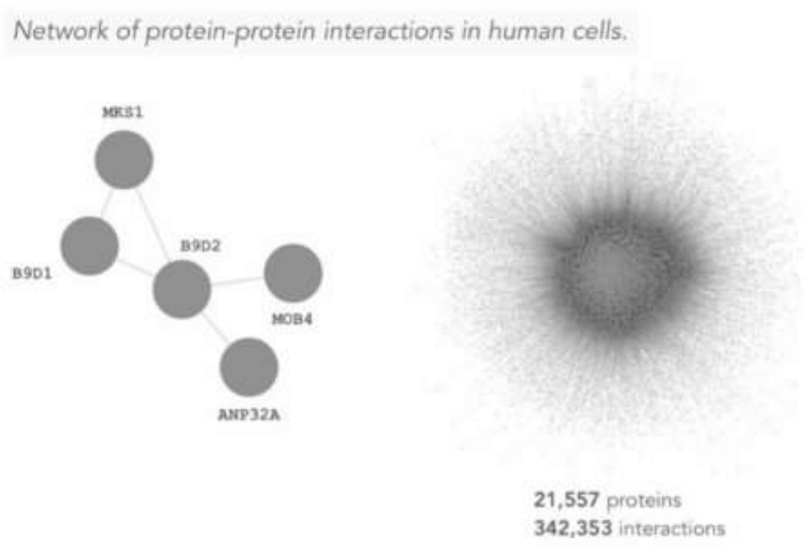
  - However, their effects do not stop at the target protein. Through the network of protein and RNA interactions, the effects spread to many other biological components.

- **Diseases**
  - These network interactions influence multiple diseases, such as:
    * Neurological disease
    * Cardiovascular disease
    * Kidney disease
  - This demonstrates that different diseases can share common molecular pathways.

Electronic Health Records

Clinical information from Electronic Health Records (EHRs) is connected to disease data, allowing researchers to relate molecular mechanisms to patient outcomes and treatments.

3.3 Why networks in Biology ?



These two images illustrate why **network analysis** is important in biology by using a **protein–protein interaction (PPI) network** in human cells.

Image 1: Protein–Protein Interaction Network

The left side of the figure shows a small **protein–protein interaction (PPI) network**, where each **node** represents a protein and each **edge** (line) represents an interaction between two proteins.

Proteins such as *MKS1*, *B9D1*, *B9D2*, *MOB4*, and *ANP32A* are connected because they physically or functionally interact with one another.

This small network is a simplified example of the much larger human protein interaction network shown on the right.

The large network consists of **21,557 proteins** connected by **342,353 interactions**, demonstrating the enormous complexity of biological systems.

The figure highlights that proteins do not function independently; instead, they operate as part of interconnected networks that regulate a wide range of cellular processes.

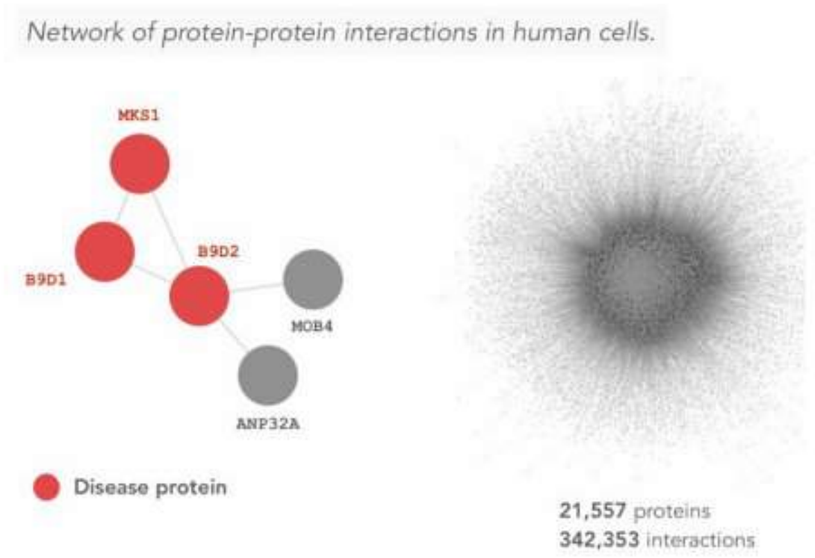


Image 2: Disease Protein Network

The second image uses the same **protein–protein interaction (PPI) network** but highlights **disease-associated proteins** in red (*MKS1*, *B9D1*, and *B9D2*), while the remaining proteins (*MOB4* and *ANP32A*) are shown in gray.

The clustering of the red proteins indicates that disease-related proteins tend to interact with one another or lie close together in the network rather than being randomly distributed.

This suggests that diseases often arise from the disruption of entire biological pathways or protein complexes rather than from a single defective protein.

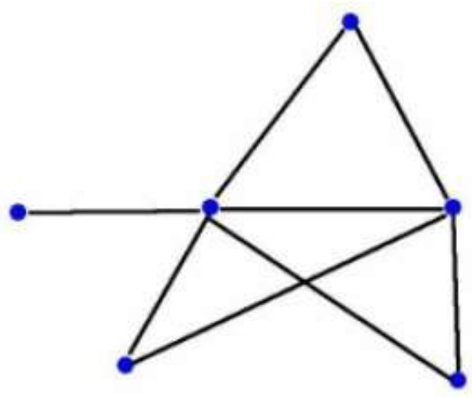
The large network on the right again represents the complete human protein interaction network, emphasizing that identifying disease-associated proteins within this network can help researchers understand disease mechanisms, identify biomarkers, and discover potential drug targets.

Conclusion

The two figures demonstrate that biological functions are governed by complex networks of interacting proteins. By analyzing these networks, scientists can identify **disease-associated proteins**, understand how diseases spread through cellular pathways, and develop more effective therapeutic strategies.

Chapter 4

The Spectral Analysis on Biology Networks



The figure illustrates the fundamental concept of a **graph** in graph theory. The upper part of the figure shows an **undirected graph** composed of six vertices (represented by blue dots) connected by edges (represented by black lines). Each vertex denotes an individual entity or object, while each edge represents a relationship or connection between two vertices. The graph includes horizontal, vertical, and diagonal edges, demonstrating that vertices can be interconnected in various ways. One vertex on the left is connected to the rest of the graph by a single edge, representing a node with a degree of one. Since every vertex can be reached from every other vertex through one or more paths, the graph is considered **connected**.

The lower part of the figure presents the formal mathematical definition

of a graph, expressed as

$$G = (V, E) \tag{4.1}$$

where:

- G denotes the graph,
  - V represents the set of **vertices (nodes)**, and
  - E represents the set of **edges**, with each edge connecting a pair of vertices.

Mathematically, a graph is defined as an ordered pair $G = (V, E)$, where V is a finite set of vertices and E is a set of two-element subsets of V. This representation provides a simple yet powerful framework for modeling relationships among entities.

Graphs are widely used to represent interconnected systems in various domains, including computer science, social networks, transportation, and biology. In computational biology, for example, vertices may represent genes, proteins, cells, or drugs, while edges represent biological interactions such as protein–protein interactions, gene regulatory relationships, or drug–target associations. This graph representation forms the foundation of **Graph Neural Networks (GNNs)**, which exploit graph structures to learn meaningful representations for tasks such as node classification, link prediction, molecular property prediction, and drug discovery. The graph is denoted by G, which is defined as an ordered pair $G = (V, E)$, where V represents the set of vertices (or nodes) and E represents the set of edges connecting pairs of vertices. Each vertex corresponds to an individual entity, while each edge represents a relationship or interaction between two vertices. Mathematically, V is a finite set of vertices, and E is a set of two-element subsets of V, indicating that every edge connects exactly two vertices. This mathematical representation provides a simple yet powerful framework for modeling relationships among entities and forms the basis for graph theory and its applications in computer science, network analysis, and computational biology.

4.0.1 Normalized Laplacian Matrix

Normalized Laplacian matrix, which is a fundamental matrix representation used in graph theory, spectral graph analysis, and Graph Neural Networks (GNNs). The normalized Laplacian matrix captures the structural properties of a graph while accounting for the degree of each vertex, making it suitable for analyzing graphs with varying node connectivity.

The normalized Laplacian matrix is denoted by

$$\Delta = [a_{ij}], \tag{4.2}$$

where each element a_{ij} is defined as

$$a_{ij} = \begin{cases} 1, & \text{if } i = j \text{ and } n_i \neq 0, \\ -\dfrac{1}{n_j}, & \text{if } i \sim j \text{ (i.e., vertices } i \text{ and } j \text{ are adjacent)}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

where:

- Δ denotes the normalized Laplacian matrix of the graph.
- a_{ij} represents the element located in the i^{th} row and j^{th} column of the matrix.
- $i = j$ indicates the diagonal entries of the matrix.
- n_i denotes the degree of vertex i, which is the number of edges incident to that vertex.
- $i \sim j$ indicates that vertices i and j are directly connected by an edge.

The diagonal elements of the normalized Laplacian matrix are assigned the value 1 for vertices having nonzero degree, while the off-diagonal elements corresponding to adjacent vertices are assigned the value $-\frac{1}{n_j}$. All remaining entries are assigned a value of 0, indicating that no direct edge exists between the corresponding pair of vertices.

The normalized Laplacian matrix plays an important role in spectral graph theory, graph clustering, community detection, graph signal processing, and Graph Neural Networks (GNNs). Its eigenvalues and eigenvectors provide valuable information about the connectivity and structural properties of a graph. In computational biology, the normalized Laplacian matrix is widely used to analyze biological networks such as protein–protein interaction networks, gene regulatory networks, and molecular graphs, enabling applications including node classification, link prediction, and drug discovery.

eigenvalue of the normalized Laplacian matrix Δ. Eigenvalues and their corresponding eigenvectors are fundamental concepts in linear algebra, spectral graph theory, and Graph Neural Networks (GNNs), as they characterize the intrinsic structural properties of a graph.

An eigenvalue λ of the normalized Laplacian matrix Δ is defined by the existence of a nonzero vector u such that

$$\Delta u = \lambda u, \tag{4.4}$$

where:

- Δ denotes the normalized Laplacian matrix of the graph.
- λ represents an eigenvalue of the matrix.
- u denotes the corresponding eigenvector, where $u \neq 0$.
- The equation $\Delta u = \lambda u$ is known as the **eigenvalue equation**, indicating that when the matrix Δ acts on the eigenvector u, the resulting vector remains in the same direction but is scaled by the factor λ.

The collection of all eigenvalues of the normalized Laplacian matrix is referred to as the **spectrum** of the graph. These eigenvalues and their corresponding eigenvectors provide important information about the graph's structural properties, including connectivity, clustering, diffusion processes, and community structure.

In Graph Neural Networks (GNNs), spectral graph convolution methods utilize the eigenvalues and eigenvectors of the normalized Laplacian matrix to perform convolution operations on irregular graph structures.

Similarly, in computational biology, spectral analysis of biological networks such as protein–protein interaction networks, gene regulatory networks, and molecular graphs facilitates the identification of functional modules, community detection, and the analysis of complex biological interactions.

- All eigenvalues are bounded in[0,2],which makes it convenient to compare different networks.
- The normalized Laplacian matrix has been studied by several authors from the view point of geometric analysis. It is ananalogous of the Laplace-Beltramioperator in Riemannian geometry.As in Riemannian case,the eigenvalues here also represent important properties of the underlying graph, such as, connectivity, bipartition.

Normalized Laplacian matrix and explains the significance of its eigenvalues in spectral graph theory. The normalized Laplacian matrix is widely used because its eigenvalues possess well-defined mathematical properties that facilitate the analysis and comparison of graph structures.

The first point states that all eigenvalues of the normalized Laplacian matrix lie within the interval $[0, 2]$. Since the eigenvalues are bounded within this fixed range, they provide a standardized measure for comparing different graphs or networks regardless of their size or degree distribution. This bounded spectrum simplifies the analysis of complex networks and enables meaningful comparisons between graphs from different domains.

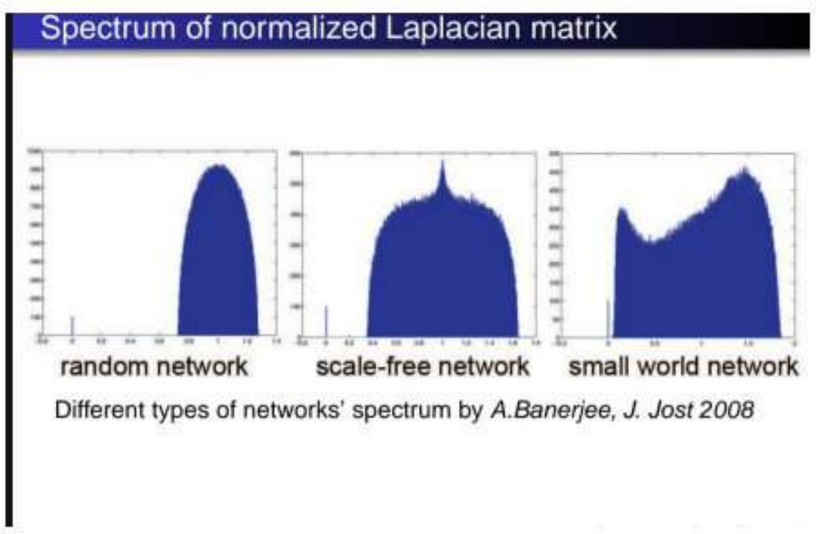
The second point emphasizes that the normalized Laplacian matrix has been extensively studied from the perspective of **geometric analysis**. It serves as the graph-theoretic analogue of the **Laplace–Beltrami operator** in **Riemannian geometry**, which describes diffusion and geometric properties on continuous manifolds. Similarly, in graph theory, the normalized Laplacian captures the structural characteristics of discrete networks.

The eigenvalues of the normalized Laplacian reveal several important properties of the underlying graph. In particular, they provide in-

formation about **graph connectivity**, **bipartiteness**, **community structure**, and the overall topology of the network. For example, the smallest eigenvalue is always zero, and its multiplicity indicates the number of connected components in the graph. Likewise, the largest eigenvalue provides insights into whether the graph is bipartite.

These spectral properties make the normalized Laplacian an essential tool in **spectral clustering**, **graph partitioning**, **Graph Neural Networks (GNNs)**, and **computational biology**. In biological network analysis, the eigenvalues and eigenvectors of the normalized Laplacian are used to identify functional modules, analyze protein–protein interaction networks, investigate gene regulatory networks, and study molecular structures. Consequently, spectral analysis based on the normalized Laplacian has become a fundamental technique for understanding the organization and behavior of complex biological systems.

4.0.2 Spectrum of Normalized Laplacian Matrix



The figure illustrates the **spectrum of the normalized Laplacian matrix** for three different types of complex networks: **random networks**, **scale-free networks**, and **small-world networks**. Each plot represents the distribution of the eigenvalues of the normalized Laplacian matrix, demonstrating how different network topologies produce distinct spectral characteristics. Since the eigenvalues of the normalized Laplacian are bounded within the interval $[0, 2]$, the spectral distributions can be directly compared across different network models.

The **left panel** shows the spectrum of a **random network**. The eigenvalues are concentrated around the middle of the spectrum, producing a relatively smooth and symmetric distribution. This behavior reflects the homogeneous structure of random graphs, where vertices tend to have similar connectivity patterns and edges are distributed uniformly.

The **middle panel** represents the spectrum of a **scale-free network**. Unlike the random network, the spectral distribution exhibits a pronounced central peak and a broader spread of eigenvalues. This characteristic arises from the heterogeneous degree distribution of scale-free networks, where a small number of highly connected hub nodes coexist with many nodes having relatively few connections. The presence of these hubs significantly influences the spectral properties of the network.

The **right panel** illustrates the spectrum of a **small-world network**. The distribution is asymmetric and exhibits multiple peaks, reflecting the combination of high local clustering and short average path lengths that characterize small-world networks. These spectral features capture the balance between local neighborhood connectivity and efficient global communication within the network.

4.1 Eigenvalue sets

Homo sapiens (5923): $\{0^{\{632\}}$, 0.003413, 0.0058298, 0.006595, 0.010495, 0.015472, 0.015935, 0.016366, 0.019801, 0.020154, 0.020775, . . . , 0.95121, 0.95259, 0.95513, 0.96187, 0.96311, 0.96845, 0.9724, 0.97805, 0.98627, 0.99116, $1^{\{2235\}}$, 1.0528, 1.0569, 1.0596, 1.0614, 1.0621, 1.0634, 1.0661, 1.0692, 1.0698, 1.0782, . . . , 1.9723, 1.9767, 1.9789, 1.9792, 1.9802, 1.984, 1.9842, 1.9843, 1.9932, 1.9935, $2^{\{567\}}\}$

Mus musculus (2154): $\{0^{\{296\}}$, 0.001047, 0.0014184, 0.001757, 0.0035151, 0.0041983, 0.0049098, 0.0069461, 0.0076774, 0.0083991, 0.010024, . . . , 0.88382, 0.88938, 0.8949, 0.90483, 0.91514, 0.92001, 0.92373, 0.92717, 0.94595, 0.96781, $1^{\{639\}}$, 1.0101, 1.0188, 1.079, 1.0961, 1.1161, 1.119, 1.1471, 1.1528, 1.1544, 1.1554, . . . , 1.9834, 1.9854, 1.9874, 1.9892, 1.9908, 1.9909, 1.9931, 1.9936, 1.9963, 1.9983, $2^{\{276\}}\}$

The figure titled **Eigenvalues Sets** presents the eigenvalue spectra of biological networks for two organisms, *Homo sapiens* (human) and *Mus musculus* (mouse). The spectra correspond to the normalized Laplacian matrices of their respective protein–protein interaction (PPI) networks. Each spectrum consists of eigenvalues arranged in ascending order, with repeated eigenvalues represented by their multiplicities. The *Homo sapiens* network contains 5923 vertices, while the *Mus musculus* network consists of 2154 vertices. The notation $0^{\{632\}}$ and $0^{\{296\}}$ indicates that the zero eigenvalue occurs 632 and 296 times in the human and mouse networks, respectively, corresponding to the number of connected components in each graph. Similarly, the large multiplicities of the eigenvalue 1 represent the presence of duplicated vertices or recurring structural motifs, which are common features of biological networks due to evolutionary duplication events. The eigenvalues are bounded within the interval $[0, 2]$, a characteristic property of the normalized Laplacian matrix. Overall, the eigenvalue spectra provide valuable insights into the connectivity, modular organization, and structural properties of the biological interaction networks.

4.1.1 Eigenvalue Distributions of Two Biological Networks:

The slide compares the eigenvalue distributions of two biological networks:

> The figure presents the eigenvalue spectra of the normalized Laplacian matrices for the protein interaction networks of *Homo sapiens* and *Mus musculus*. The total number of proteins (nodes) in each network is given in parentheses: 5923 for *Homo sapiens* and 2154 for *Mus musculus*. The eigenvalues lie in the interval $[0, 2]$, which is characteristic of the normalized Laplacian matrix. The notation 0^{632}, 1^{2235}, and 2^{567} for *Homo sapiens*, and 0^{296}, 1^{639}, and 2^{276} for *Mus musculus*, denotes that the eigenvalues 0, 1, and 2 occur with multiplicities of 632, 2235, and 567 for the human network, and 296, 639, and 276 for the mouse network, respectively. The remaining eigenvalues are arranged in ascending order,

with only representative values shown and the omitted values indicated by ellipses. Compared to the mouse network, the human protein interaction network contains more nodes and exhibits a richer eigenvalue spectrum. The multiplicities of the eigenvalues provide important information about the structural properties of the networks, including connected components, duplicated structures, and bipartite-like subgraphs. Thus, the eigenvalue spectra serve as an effective tool for characterizing and comparing the topological complexity of biological networks.

4.2 Spectrum Plot

A spectrum plot is a graphical representation of the eigenvalue distribution of a network. Instead of displaying individual eigenvalues, a Gaussian kernel is used to smooth the eigenvalue set into a continuous spectral density function. This provides a clearer visualization of the network's structural characteristics and facilitates comparison between different networks. For a network G with n nodes, let the set of eigenvalues of its normalized Laplacian matrix be

$$\{\lambda_i\}_{i=1}^{n}.$$

The spectral density function is obtained using the Gaussian kernel as

$$\rho(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sqrt{2\pi\sigma_G^2}}\exp\left(-\frac{(x-\lambda_i)^2}{2\sigma_G^2}\right),$$

where

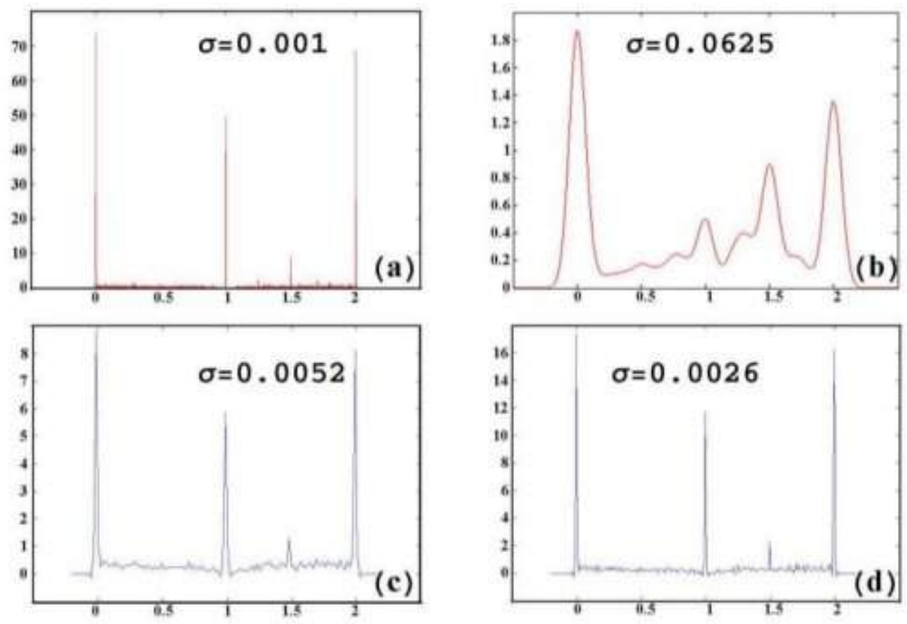
* n denotes the total number of nodes in the network.
* λ_i represents the i-th eigenvalue of the normalized Laplacian matrix.
* $\rho(x)$ denotes the estimated spectral density function.
* σ_G is the bandwidth (or smoothing factor) of the Gaussian kernel.

The equation shows that each eigenvalue contributes a Gaussian curve centered at its corresponding eigenvalue. The summation of all these Gaussian kernels produces a smooth approximation

of the eigenvalue distribution, known as the *spectrum plot*. This continuous representation provides a clearer visualization of the spectral characteristics of the network and facilitates the comparison of different network structures.

The bandwidth parameter, denoted by σ_G, plays a crucial role in the Gaussian kernel density estimation. If the bandwidth is chosen to be too small, the resulting spectrum contains numerous sharp peaks and becomes highly sensitive to small variations in the eigenvalues, making the spectrum appear noisy. On the other hand, if the bandwidth is too large, excessive smoothing occurs, causing important structural features and distinctive spectral characteristics of the network to be lost. Therefore, an appropriate choice of the bandwidth σ_G is essential to achieve a balance between preserving meaningful structural information and reducing noise in the estimated spectral density.

4.2.1 Spectrum Plot Graph



In this figure illustrates the kernel density estimation (KDE) of the eigenvalue spectrum using different values of the bandwidth parameter, σ. The bandwidth controls the degree of smoothing applied to the eigenvalue distribution.

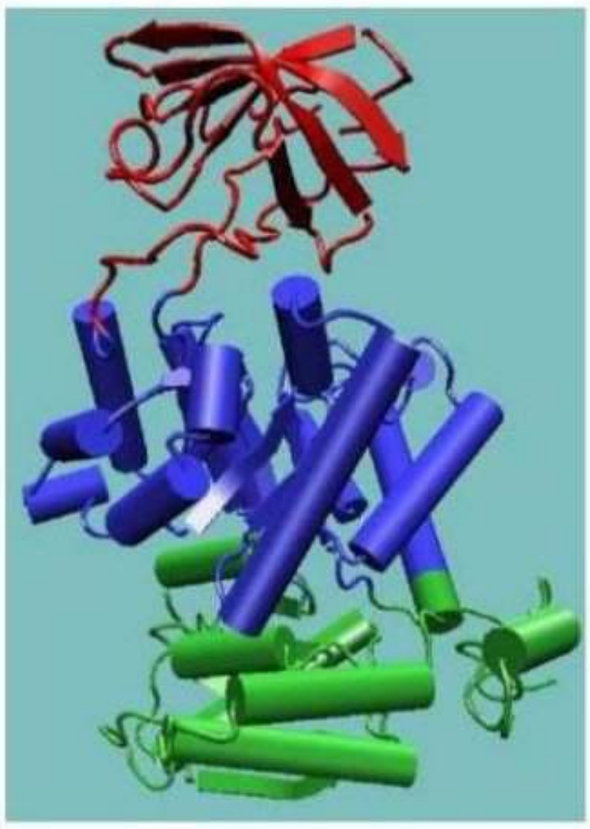
* **(a)** $\sigma = 0.001$**:** A very small bandwidth produces an *under-*

smoothed spectrum. The distribution consists of extremely sharp and narrow peaks at the eigenvalues, making the spectrum noisy and discontinuous. Although the exact eigenvalue locations are preserved, the overall distribution is difficult to interpret.

* **(b)** $\sigma = 0.0625$**:** A much larger bandwidth results in an *oversmoothed* spectrum. The sharp peaks merge into broad, smooth curves, causing important spectral features and fine details to disappear. Consequently, the spectrum appears overly generalized.
* **(c)** $\sigma = 0.0052$**:** This bandwidth provides a balanced level of smoothing. The major spectral peaks remain distinct while random fluctuations are reduced. The resulting spectrum accurately represents the underlying eigenvalue distribution without excessive noise.
* **(d)** $\sigma = 0.0026$**:** This spectrum also preserves the important peaks with only slight smoothing. Compared with (a), the noise is significantly reduced, while compared with (b), the essential spectral features remain clearly visible. This bandwidth provides a clear representation of the eigenvalue distribution.

The figure demonstrates the importance of selecting an appropriate bandwidth for kernel density estimation. A very small bandwidth produces a noisy spectrum with highly localized peaks, whereas a very large bandwidth oversmooths the distribution and obscures important structural information. Intermediate bandwidths, as shown in (c) and (d), provide an effective balance by preserving significant spectral characteristics while reducing noise, making them more suitable for spectral analysis of networks.

4.3 Protein and Domain



The figure shows the three-dimensional structure of a protein, with different domains represented in different colors. Proteins are biological molecules made up of amino acids that fold into specific three-dimensional shapes to perform various functions in living organisms.

* **Red region (top):**
  - Represents the first protein domain.
  - It is a compact part of the protein with its own unique structure and function.
  - This domain may interact with other proteins or molecules.
* **Blue region (middle):**
  - Represents the second protein domain.
  - It consists mainly of *alpha-helices* (spiral-shaped structures).
  - This domain contributes to the protein's stability and bi-

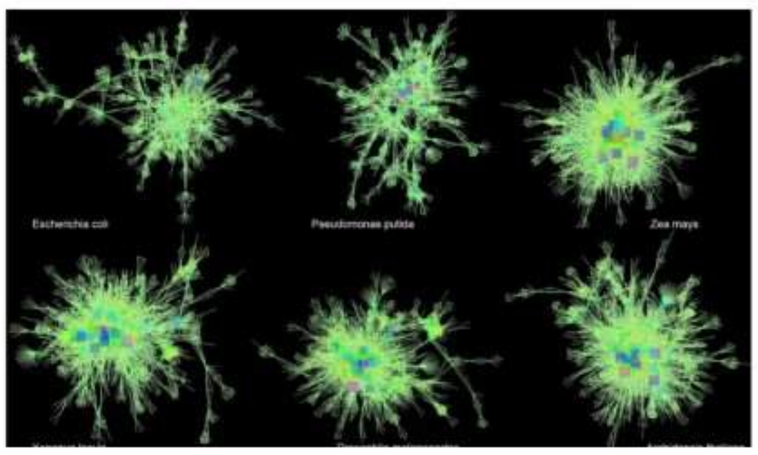
ological activity.

  * **Green region (bottom):**
    - Represents the third protein domain.
    - Like the blue region, it contains several *alpha-helices* but forms a separate functional unit.
    - It may participate in binding or catalytic functions depending on the protein.

A **protein domain** is a distinct part of a protein that has its own structure and function. Each domain can fold independently and often carries out a specific biological task. Many proteins contain multiple domains that work together, allowing the protein to perform complex functions.

The figure illustrates that proteins are composed of one or more structural domains, each contributing to the protein's stability and biological activity.

4.4 Domain Co-occurence Networks



The figure shows **Domain Co-occurrence Networks (DCNs)** for six different organisms, including *Escherichia coli*, *Pseudomonas putida*, *Zea mays*, *Xenopus laevis*, *Drosophila melanogaster*, and *Arabidopsis thaliana*. Each network illustrates how different protein domains co-occur within proteins of an organism.

  * Nodes (vertices) represent **protein domains**.
  * Edges (links) connect two domains that appear together in the same protein.

* The green web-like structure represents the overall domain co-occurrence network.
* The dense central region contains highly connected domains that frequently co-occur with many other domains.
* The outer branches (peripheral nodes) represent specialized or rarely occurring domains with fewer connections.
* The colored square nodes (blue, purple, pink, etc.) highlight important or highly connected domains (hub domains) that play significant roles in maintaining the connectivity of the network.
* Although the network structures differ among organisms, each exhibits a densely connected core surrounded by sparsely connected peripheral domains, reflecting both conserved and organism-specific domain organizations.

4.5 How the different Organisms act in Domain Co-occurrence Networks (DCNs)

?

Table 4.1: Comparison of Domain Co-occurrence Networks across Different Organisms

Organism	**Network Complexity**	**Major Biological Functions**
Escherichia coli	Low	Basic metabolism, DNA replication
Pseudomonas putida	Moderate	Environmental adaptation, biodegradation
Zea mays	High	Photosynthesis, growth, stress response
Xenopus laevis	High	Development, immunity, cell differentiation
Drosophila melanogaster	High	Gene regulation, development, signaling
Arabidopsis thaliana	High	Photosynthesis, hormone signaling, flowerin

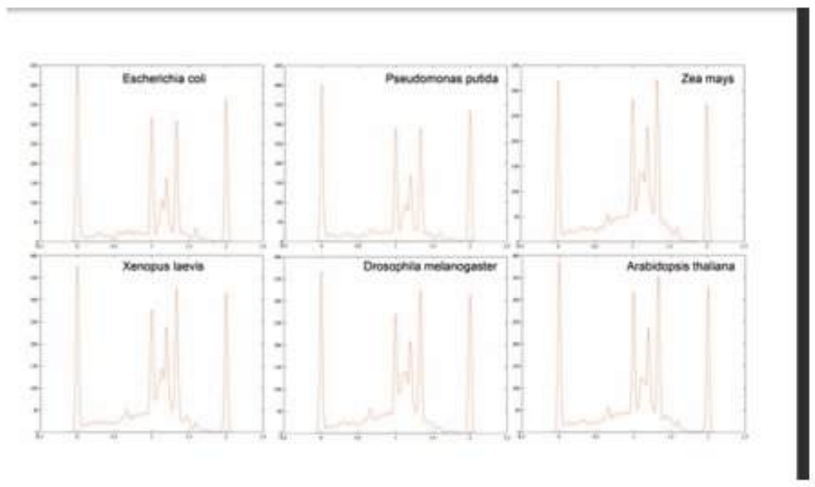
Roles of Domain Co-occurrence Networks (DCNs)

* Study protein architecture by revealing how different domains combine to form multidomain proteins.
* Understand protein evolution through conserved domain combinations across species.

* Predict protein function by identifying domains that frequently occur together.
* Identify hub domains that are essential for maintaining network connectivity and biological functions.

The figure demonstrates that proteins are constructed from combinations of functional domains rather than acting as isolated units. Domain Co-occurrence Networks help researchers understand protein evolution, identify key functional domains, predict protein functions, compare organisms, and investigate disease mechanisms.

4.6 The Spectrums of Domain Co-occurence Networks



The figure presents the **kernel density estimation (KDE) spectrum plots** of the eigenvalue distributions for the **Domain Co-occurrence Networks (DCNs)** of six different organisms: *Escherichia coli*, *Pseudomonas putida*, *Zea mays*, *Xenopus laevis*, *Drosophila melanogaster*, and *Arabidopsis thaliana*. Each subplot represents the spectral density of the normalized Laplacian eigenvalues for one organism.

* ***Escherichia coli*:** The spectrum exhibits several sharp peaks, with a dominant peak near the lower eigenvalues and additional peaks at higher eigenvalues. The relatively simple distribution reflects the lower complexity of its domain co-occurrence network.
* ***Pseudomonas putida*:** The spectrum contains multiple

distinct peaks with a broader distribution in the middle region, indicating a more diverse network organization and greater environmental adaptability.

* ***Zea mays*:** The spectrum is broader and contains several closely spaced peaks, suggesting a highly interconnected and functionally diverse network. The increased spectral complexity reflects the larger number of interacting protein domains.
* ***Xenopus laevis*:** The spectrum displays multiple pronounced peaks over a wider range of eigenvalues, indicating a complex domain organization associated with vertebrate development, immune responses, and cell differentiation.
* ***Drosophila melanogaster*:** The spectral density shows several major peaks together with intermediate fluctuations, reflecting a well-organized network containing numerous hub domains involved in gene regulation and developmental pathways.
* ***Arabidopsis thaliana*:** The spectrum resembles that of *Zea mays*, with multiple dominant peaks and a broad central distribution. This indicates a dense and highly connected domain co-occurrence network supporting photosynthesis, hormone signaling, and stress-response mechanisms.

The spectrum plots demonstrate that each organism possesses a unique eigenvalue distribution, reflecting differences in the topology of its Domain Co-occurrence Network. Simpler organisms, such as *Escherichia coli* and *Pseudomonas putida*, exhibit relatively simple spectral patterns corresponding to less complex domain interactions. In contrast, higher organisms, including *Zea mays*, *Xenopus laevis*, *Drosophila melanogaster*, and *Arabidopsis thaliana*, display broader and more intricate spectra with multiple prominent peaks, indicating denser connectivity, greater structural complexity, and increased functional diversity. These spectral characteristics provide valuable insights into the organization, evolution, and biological functions of protein domains across different species.

4.7 Protein–Protein Interaction Networks of Six Organisms

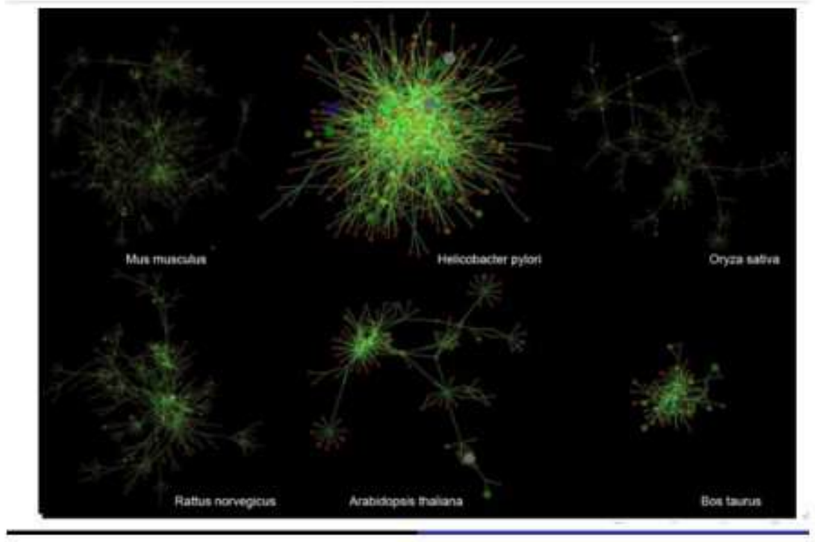


Figure compares the protein–protein interaction (PPI) networks of six different organisms. In these networks, each **node** represents a protein, while each **edge** represents an interaction between two proteins. The variation in network size, density, and connectivity reflects the biological complexity and functional organization of each organism.

Table 4.2: Role of Different Organisms in Protein–Protein Interaction Networks

Organism	**Role in Protein Interaction Network**
Mus musculus (Mouse)	A mammalian model organism with a complex PPI network used to study human diseases, development, and cellular processes.
Helicobacter pylori	A bacterium with a densely connected PPI network that helps researchers understand bacterial survival, metabolism, and pathogenic mechanisms.
Oryza sativa (Rice)	A crop plant whose PPI network is used to investigate plant growth, stress tolerance, and agriculturally important traits.
Rattus norvegicus (Rat)	An important mammalian model with a large PPI network for studying physiology, neuroscience, and drug responses.
Arabidopsis thaliana	A model plant with a well-characterized PPI network used to study plant signaling pathways, development, and gene regulation.
Bos taurus (Cow)	A livestock species whose PPI network is used to study reproduction, immunity, metabolism, and economically important traits.

The figure demonstrates that protein–protein interaction networks differ in size, density, and connectivity among organisms. These networks help identify key hub proteins, understand biological pathways, predict protein functions, and compare molecular mechanisms across different species.

4.7.1 The Spectrums of Protein Interaction Networks

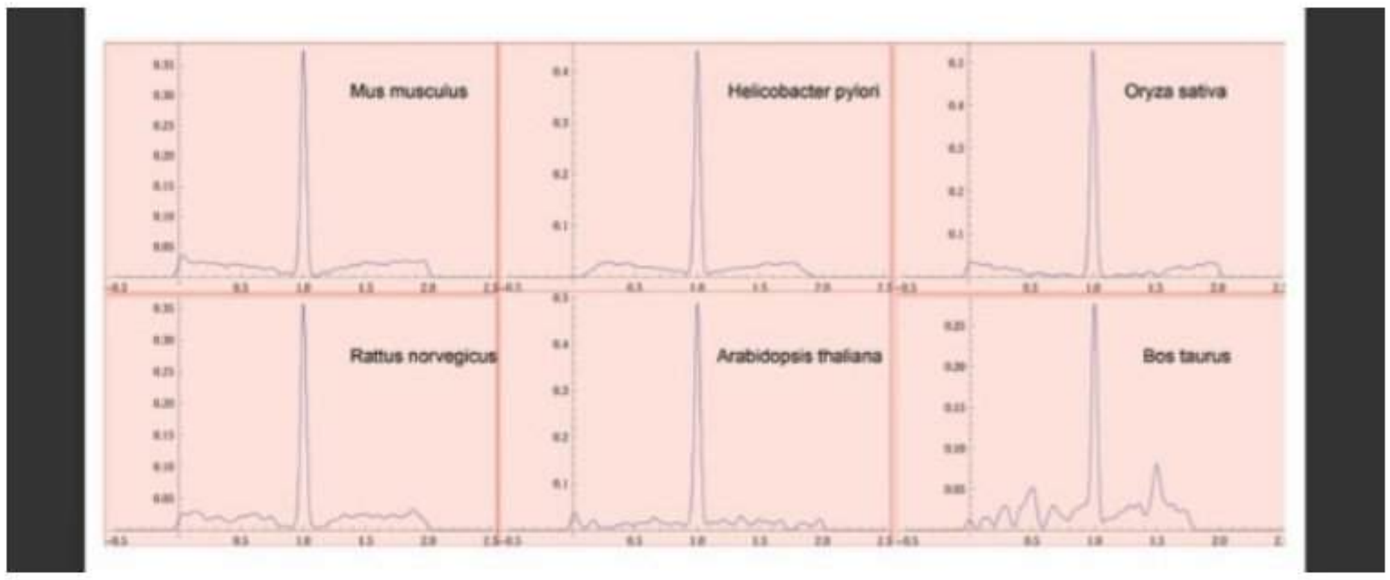


Figure compares the distributions of six different species: *Mus musculus*, *Helicobacter pylori*, *Oryza sativa*, *Rattus norvegicus*, *Arabidopsis thaliana*, and *Bos taurus*. All six plots exhibit a prominent peak at approximately $x = 1.0$, indicating a common dominant feature across the species. The distributions for *Mus musculus*, *Helicobacter pylori*, *Oryza sativa*, *Rattus norvegicus*, and *Arabidopsis thaliana* display a single sharp peak with minimal background fluctuations. In contrast, the *Bos taurus* distribution shows additional fluctuations before and after the main peak, suggesting comparatively higher variability. Overall, the figure demonstrates a similar distribution pattern among the species, with only minor differences in the background signals.

4.7.2 Role of Spectral Graphs in Studying Protein Interactions

Spectral graphs are used to characterize proteins and compare their molecular properties. They provide valuable insights into protein structure and interactions through the analysis of spectral features.

* **Identify Protein Signatures**
    - Each protein or protein sample produces a characteristic spectrum.

- The position and shape of spectral peaks provide information about the protein's chemical environment.

* **Detect Protein–Protein Interactions**
  - When two proteins interact, their spectra may change.
  - Researchers analyze the following spectral changes:
  - Peak shifts
  - Changes in peak intensity
  - Peak broadening or disappearance
  - These spectral changes indicate protein binding events or conformational changes in the protein structure.

4.8 Binary Tree Reconstruction

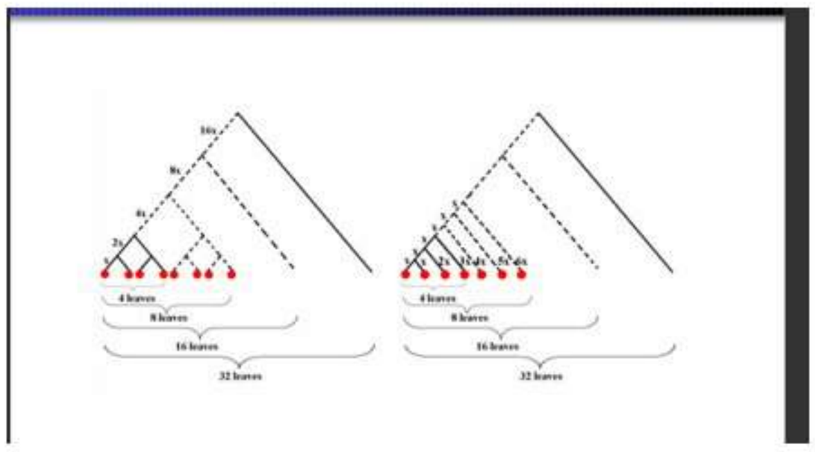


Figure illustrates the recursive reconstruction of a binary tree using a hierarchical divide-and-conquer approach. The figure compares two reconstruction strategies for building a binary tree containing 32 leaf nodes.
The **red circles** at the bottom of each diagram represent the leaf nodes, while the solid black lines denote the final reconstructed binary tree. The dashed lines indicate the intermediate recursive partitions used during the reconstruction process. The brackets beneath each tree represent the recursive grouping of leaf nodes into subtrees consisting of 4, 8, 16, and 32 leaves.

Balanced Reconstruction

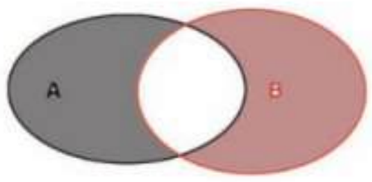
In the left diagram, the reconstruction follows a balanced recursive strategy. Initially, groups of four leaf nodes are combined to form small subtrees. These subtrees are then recursively merged into larger groups of 8, 16, and finally 32 leaf nodes. The labels *2x*, *4x*, *8x*, and *16x* represent the computational effort or recursive expansion at each level of the tree.

Incremental Reconstruction

The right diagram illustrates an incremental reconstruction strategy. Instead of immediately combining equally sized groups, several local reconstruction operations are performed on smaller subsets of leaf nodes before merging them into larger subtrees. The labels *x*, *2x*, *3x*, *4x*, *5x*, and *6x* indicate the cumulative computational cost associated with these repeated reconstruction steps.

Both reconstruction strategies ultimately generate a binary tree containing 32 leaf nodes. However, they differ in the order and number of recursive operations required during reconstruction. The balanced strategy performs uniform recursive merging at each level, whereas the incremental strategy performs additional local reconstruction steps before combining larger subtrees. This comparison demonstrates how different reconstruction approaches influence computational complexity and reconstruction efficiency while producing the same hierarchical tree structure.

4.9 Edit Distance



The figure presents two overlapping sets, denoted as Graph A and Graph B, using a Venn diagram. The left set (A) is shaded in gray, while the right set (B) is shaded in light red. The overlapping region represents the edges that are common to both graphs, whereas the non-overlapping regions correspond to the edges that are unique to each graph.

The graph edit distance, denoted by d_e, is defined as the number (or ratio) of different edges between two graphs. It represents the minimum number of edge insertion, deletion, or substitution operations required to transform Graph A into Graph B.

A larger overlapping region between the two graphs indicates a smaller edit distance, implying that the graphs are structurally more similar. Conversely, a smaller overlap and larger non-overlapping regions correspond to a greater edit distance, reflecting increased structural differences between the graphs.

Graph edit distance is widely used in applications such as protein interaction networks to compare network structures, identify conserved interaction patterns, measure structural similarity between biological networks, and detect evolutionary changes across different organisms. It serves as an effective metric for quantifying the degree of similarity or dissimilarity between complex graph-based representations.

Overall, the figure demonstrates that graph similarity is determined by the proportion of shared edges, while graph differences are quantified by the edit distance based on the edges that are present in one graph but absent in the other. The figure contains four line plots comparing the **Similarity Ratio** (y-axis) against X (x-axis) for different numbers of leaves: 4, 8, 16, and 32.

Edit Distance

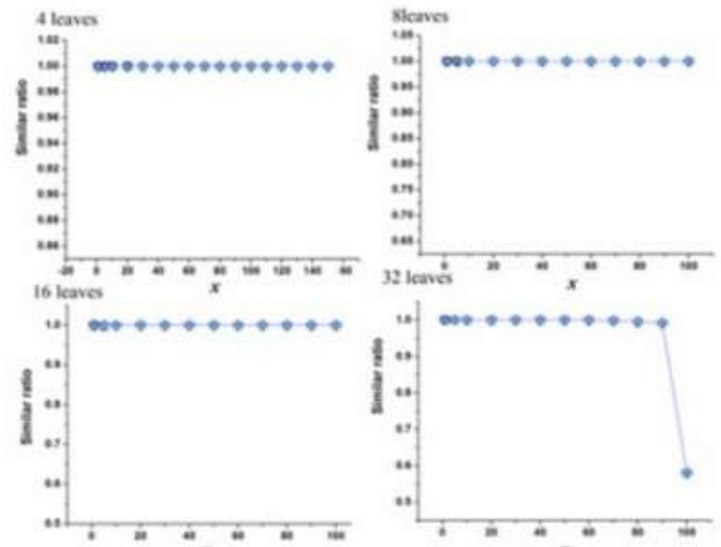


Figure 4.1: Similarity ratio for different numbers of leaves.

* **4 leaves:** The similarity ratio remains constant at approximately 1.0 across all values of X, indicating perfect or near-perfect similarity regardless of X.
* **8 leaves:** The similarity ratio is also consistently close to 1.0 for all values of X, demonstrating stable and high similarity.
* **16 leaves:** Similar to the previous cases, the similarity ratio remains around 1.0 throughout the entire range of X, with no noticeable variation.
* **32 leaves:** The similarity ratio stays close to 1.0 for most values of X but drops sharply to approximately 0.58 at the highest value of X (around 100), indicating a significant reduction in similarity only at the final observation.

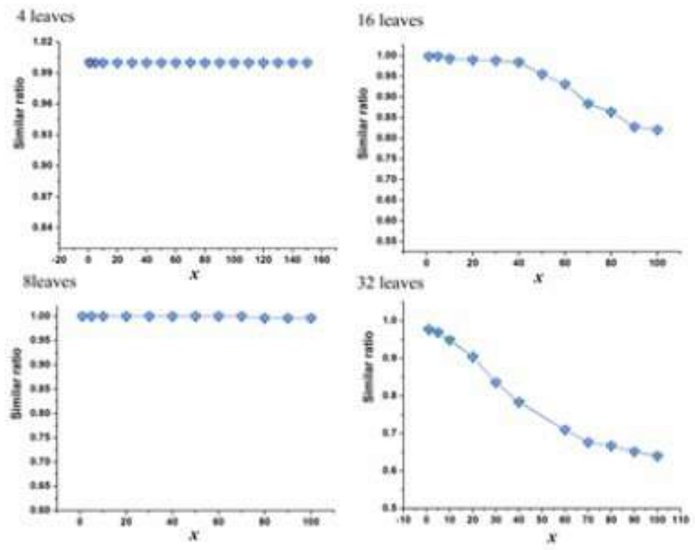


Figure 4.2: Similarity ratio for different numbers of leaves.

The results indicate that increasing the number of leaves from 4 to 16 has no noticeable effect on the similarity ratio, which remains consistently high across all values of X. For the case of 32 leaves, the similarity ratio remains stable for most values of X but exhibits a sharp decline at the largest value of X. This suggests that the system maintains robust performance for smaller tree sizes and moderate values of X, but its performance or similarity deteriorates under the combination of the highest tree complexity (32 leaves) and the maximum value of X.

Comparison of the Two Figures.

Both figures (figure 3.1 and figure 3.2) present the **Similarity Ratio** as a function of X for four different numbers of leaves (4, 8, 16, and 32). However, they exhibit noticeable differences in the behavior of the similarity ratio, particularly for the larger tree sizes.

Table 4.3: Comparison of Similarity Ratio Trends

Leaves	**First Figure**	**Second Figure**
4	Similarity ratio remains constant at approximately 1.0 for all values of X.	Similarity ratio remains constant at approximately 1.0 for all values of X.
8	Similarity ratio remains close to 1.0 throughout the range of X.	Similarity ratio also remains close to 1.0 across all values of X.
16	Similarity ratio gradually decreases from approximately 1.0 to about 0.82 as X increases.	Similarity ratio remains nearly constant at 1.0 across the entire range of X.
32	Similarity ratio decreases steadily from approximately 1.0 to about 0.64 with increasing X.	Similarity ratio remains close to 1.0 until the final value of X, where it drops sharply to approximately 0.58.

Eigenvalue Sets

Homo sapiens (5923): {$0^{\{632\}}$, 0.003413, 0.0058298, 0.006595, 0.010495, 0.015472, 0.015935, 0.016366, 0.019801, 0.020154, 0.020775, . . . , 0.95121, 0.95259, 0.95513, 0.96187, 0.96311, 0.96845, 0.9724, 0.97805, 0.98627, 0.99116, $1^{\{2235\}}$, 1.0528, 1.0569, 1.0596, 1.0614, 1.0621, 1.0634, 1.0661, 1.0692, 1.0698, 1.0782, . . . , 1.9723, 1.9767, 1.9789, 1.9792, 1.9802, 1.984, 1.9842, 1.9843, 1.9932, 1.9935, $2^{\{567\}}$}

Mus musculus (2154): {$0^{\{296\}}$, 0.001047, 0.0014184, 0.001757, 0.0035151, 0.0041983, 0.0049098, 0.0069461, 0.0076774, 0.0083991, 0.010024, . . . , 0.88382, 0.88938, 0.8949, 0.90483, 0.91514, 0.92001, 0.92373, 0.92717, 0.94595, 0.96781, $1^{\{639\}}$, 1.0101, 1.0188, 1.079, 1.0961, 1.1161, 1.119, 1.1471, 1.1528, 1.1544, 1.1554, . . . , 1.9834, 1.9854, 1.9874, 1.9892, 1.9908, 1.9909, 1.9931, 1.9936, 1.9963, 1.9983, $2^{\{276\}}$}

Figure shows a **Eigenvalues sets**. The window displays the computed eigenvalue spectra for two biological species: *Homo sapiens* and *Mus musculus*.

Homo sapiens (5923)

The first section presents a set of 5,923 eigenvalues enclosed in braces. The eigenvalues begin as

$$0^{(632)},\ 0.003413,\ 0.0058298,\ 0.006595,\ 0.010495,\ 0.015472,\ 0.015935,\ \ldots$$

where the notation $0^{(632)}$ denotes that the eigenvalue 0 occurs with multiplicity 632.

The sequence continues in increasing order and ends approximately as

$$\ldots,\ 1.9932,\ 1.9935,\ 2^{(567)},$$

indicating that the eigenvalue 2 has multiplicity 567.

Mus musculus (2154)

The second section displays a total of 2,154 eigenvalues. The sequence begins with

$$0^{(296)},\ 0.001047,\ 0.001184,\ 0.001757,\ 0.0035151,\ \ldots$$

where $0^{(296)}$ indicates that the eigenvalue 0 occurs with multiplicity 296.

The eigenvalues increase monotonically and terminate approximately as

$$\ldots,\ 1.9931,\ 1.9936,\ 1.9983,\ 2^{(276)},$$

showing that the eigenvalue 2 occurs with multiplicity 276.

The image appears to originate from a scientific or mathematical software package that computes the eigenvalue spectra of matrices associated with biological datasets for *Homo sapiens* and *Mus musculus*. The eigenvalues are confined to the interval $[0, 2]$, with a significant number

of repeated eigenvalues at 0 and 2. This spectral distribution is characteristic of normalized graph Laplacian matrices or similar graph-based network representations commonly used in biological network analysis.

4.10 Bipartite Matching Distance

For two sequences

$$X = \{x_1, x_2, x_3, \ldots, x_m\},$$

and

$$Y = \{y_1, y_2, y_3, \ldots, y_n\},$$

the bipartite matching distance is defined as

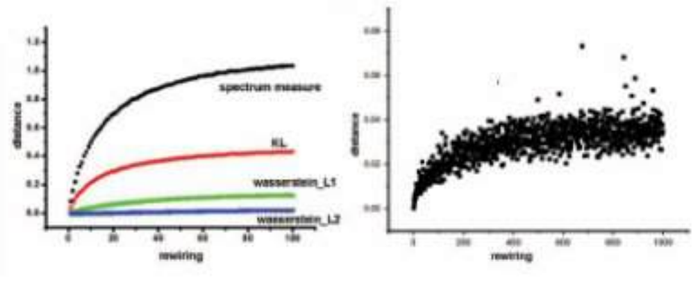
$$d_B(X, Y) = \sum_{i=1}^{m} \min_j |x_i - y_j| + \sum_{j=1}^{n} \min_i |y_j - x_i|. \tag{4.5}$$

The distance is computed in two parts:

1. For each element x_i in set X, find the closest element in set Y, compute the absolute difference, and sum these minimum distances.
2. For each element y_j in set Y, find the closest element in set X, compute the absolute difference, and sum these minimum distances.

The bipartite matching distance is the sum of these two quantities. A smaller value indicates greater similarity between the two sequences, while a larger value indicates greater dissimilarity. The distance is computed in two parts.

Distance with Random Graph



The image is a presentation slide titled **"Distance with Random Graph"**. It compares how different graph distance measures behave as the amount of graph rewiring increases. The slide contains two graphs, each presenting different types of information.

Description of the Left Graph

The left graph is a line plot comparing four graph distance measures as the number of graph rewiring operations increases.

* **X-axis:** Rewiring (number of edge rewiring operations).
* **Y-axis:** Distance.

The graph compares the following four distance measures:

* **Spectrum Measure (black curve):** The spectrum measure increases rapidly with the number of rewiring operations and attains the highest distance values, indicating that the spectral distance is highly sensitive to graph rewiring.
* **Resistance–Laplacian (RL) Distance (red curve):** The RL distance exhibits a moderate increase with rewiring and eventually stabilizes below the spectrum measure.
* **Wasserstein L_1 Distance (green curve):** The Wasserstein L_1 distance increases gradually and remains relatively small throughout the rewiring process.

* **Wasserstein L_2 Distance (blue curve):** The Wasserstein L_2 distance shows the smallest increase among all the distance measures, indicating that it is the least sensitive to graph rewiring.

The curves gradually level off as the number of rewiring operations increases, suggesting that the graph distance measures eventually stabilize after sufficient graph rewiring.

Description of the Right Graph

The right graph is a scatter plot illustrating the relationship between the number of graph rewiring operations and the corresponding graph distance.

* **X-axis:** Rewiring (number of edge rewiring operations).
* **Y-axis:** Distance.

Each black dot represents the distance obtained for a randomly rewired graph after a specific number of rewiring operations.

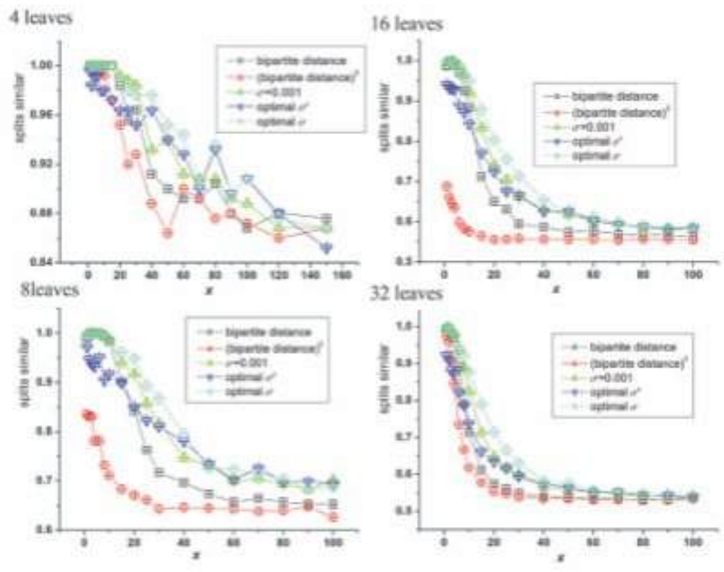
Unlike the left graph, which displays smooth curves representing the behavior of different graph distance measures, the right graph consists of scattered data points. These points illustrate the variability in the measured distances across different random graph realizations.

The scatter plot reveals the following observations:

* The average graph distance generally increases as the number of rewiring operations increases.
* There is noticeable variation in the distance values among different randomly rewired graphs.
* A few outliers exhibit significantly larger distance values than the majority of the observations, indicating occasional large deviations caused by random rewiring.

Experimental Results of Graph Distance Measures

4.10.1 Experimental Results



This figure presents the experimental comparison of several graph distance measures for trees of different sizes (4, 8, 16, and 32 leaves). Each subplot illustrates how the similarity between two graphs changes as the parameter x, representing the amount of modification or perturbation applied to the graph, increases. The experimental results are evaluated using the split similarity measure.

The x-axis represents the parameter x, which controls the experiment, such as the network size, iteration step, or distance threshold. The y-axis represents the split similarity, where a higher value indicates greater similarity between the generated and compared network splits.

1. **4 leaves network**

   The split similarity starts from a very high value (approximately 1.0). As the value of x increases, all methods show a gradual decrease in split similarity. The *bipartite distance** method (red curve) decreases faster, indicating lower similarity compared with other approaches. The optimal methods, σ and σ' (green and blue curves), maintain higher similarity

values and show better performance.

2. **8 leaves network**
   A similar decreasing trend is observed for the 8 leaves network. The *bipartite distance** curve decreases rapidly and reaches the lowest similarity value. The optimal spectral methods (σ and σ') remain more stable and achieve better similarity preservation compared with other methods.
3. **16 leaves network**
   For the 16 leaves network, the split similarity decreases smoothly from approximately 1.0 to around 0.6. As x increases, the different curves gradually converge. The optimal spectral-based methods demonstrate better preservation of split similarity compared with the bipartite distance approaches.
4. **32 leaves network**
   The 32 leaves network shows a sharper initial decrease in split similarity. After approximately $x = 40$, the curves become almost stable. The *bipartite distance** method again produces the lowest similarity values, while the optimal methods maintain higher similarity throughout the experiment.

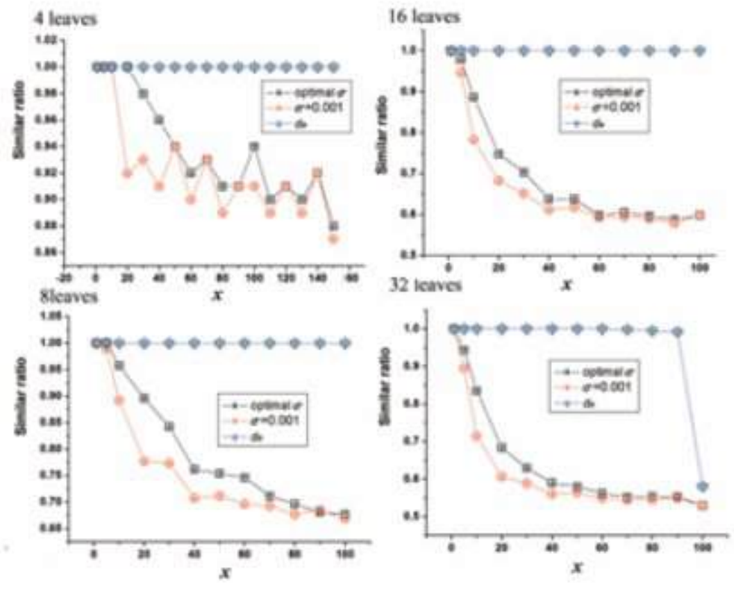
Each graph contains multiple curves corresponding to different methods:

* *Bipartite distance* (black/gray curve)
* *Bipartite distance** (red curve)
* *Optimal* σ (green curve)
* *Optimal* σ' (blue curve)

The graph demonstrates that as the network complexity increases (from $4 \rightarrow 32$ leaves), the split similarity decreases, indicating that the methods face more difficulty in preserving the original network structure.

The optimal spectral methods, σ and σ', generally achieve better performance compared with the basic bipartite distance approaches. This is because they maintain higher split similarity values across different network sizes, showing better preservation of network structure.

4.10.2 Experimental Results



The graph presents the experimental comparison of the similarity ratio for different network sizes using various methods. The figure consists of four subgraphs representing different network complexities: 4 leaves, 8 leaves, 16 leaves, and 32 leaves.

Each subgraph illustrates the variation of the similarity ratio under different experimental conditions and compares the performance of the evaluated methods.

The x-axis represents the experimental parameter x, which may correspond to the iteration number, distance threshold, or another network-related parameter. The y-axis represents the similarity ratio, which measures how effectively each method preserves the original network structure. A higher similarity ratio indicates better preservation of the network structure.

The curves compare three different approaches:

* *Optimal* σ (blue curve)
* $\sigma = 0.001$ (red curve)
* d_0 / distance-based method (black/gray curve)

1. **4 leaves network**

   The similarity ratio starts close to 1.0 for all methods. The optimal σ method remains almost constant at the maximum similarity value. The $\sigma = 0.001$ and distance-based methods decrease as x increases. The red curve shows larger

fluctuations and lower similarity compared with the optimal method.

2. **8 leaves network**

   The similarity ratio gradually decreases with increasing x. The optimal σ method maintains a stable value close to 1.0. The $\sigma = 0.001$ method decreases faster and reaches lower similarity values. The distance-based method shows a gradual decline.

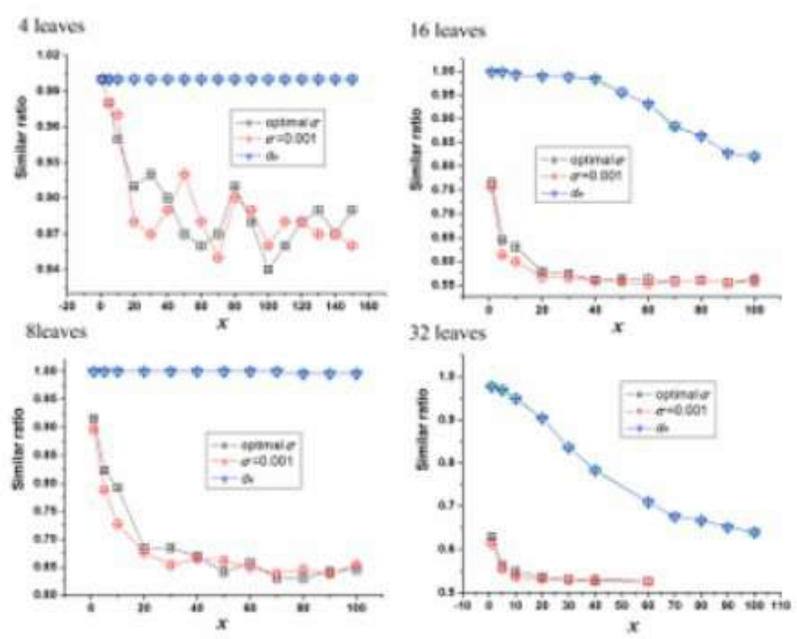
3. **16 leaves network**

   A similar decreasing pattern is observed for the 16 leaves network. The optimal σ curve remains nearly unchanged with a high similarity ratio. The other methods decrease smoothly from approximately 1.0 to around 0.6. The optimal method provides better preservation of network similarity.

4. **32 leaves network**

   The initial similarity ratio decreases sharply for the non-optimal methods. After approximately $x = 40$, the curves become relatively stable. The optimal σ method maintains a high similarity ratio throughout the experiment, demonstrating superior performance.

4.10.3 Experimental Results



The graph shows the experimental comparison of the similarity

ratio for different network sizes using three different methods. The figure consists of four subgraphs representing different network complexities: 4 leaves, 8 leaves, 16 leaves, and 32 leaves.

The x-axis represents the experimental parameter x, which may correspond to the iteration number, distance threshold, or another network-related parameter. The y-axis represents the similarity ratio, which indicates how effectively each method preserves the original network structure. A higher similarity ratio represents better preservation of the network structure.

The curves compare three different approaches:

* Optimal σ (black/gray curve)
* $\sigma = 0.001$ (red curve)
* d_0 / distance-based method (blue curve)

1. **4 leaves network**

   The similarity ratio starts close to 1.0 for all methods. The d_0 distance-based method (blue curve) remains almost constant at a high similarity value. The optimal σ method gradually decreases with increasing x. The $\sigma = 0.001$ method (red curve) shows larger fluctuations and lower similarity compared with the other methods.

2. **8 leaves network**

   The similarity ratio decreases as x increases. The d_0 method maintains a stable similarity ratio close to 1.0. The optimal σ and $\sigma = 0.001$ methods decrease significantly. The red curve reaches lower similarity values, indicating weaker preservation of the network structure.

3. **16 leaves network**

   A gradual decrease in similarity ratio is observed. The d_0 method decreases slowly but still maintains higher similarity compared with the other approaches. The optimal σ and $\sigma = 0.001$ methods drop quickly and stabilize at lower similarity values. The distance-based approach provides better performance for this network size.

4. **32 leaves network**

   The similarity ratio decreases more noticeably for all methods. The d_0 method shows a smooth decline but remains the highest-performing method. The optimal σ and $\sigma = 0.001$ methods reach lower similarity values and become stable after a certain point.

The three experimental graphs together demonstrate the performance comparison of different spectral and distance-based methods for preserving the structure of biological networks with increasing network complexity (from 4 leaves to 32 leaves). The results show that as the network size increases, the similarity ratio and split similarity generally decrease, indicating that maintaining the original network structure becomes more challenging for larger networks. The first graph shows that the optimal spectral methods (σ and σ') generally maintain higher split similarity values compared with the basic bipartite distance approaches, demonstrating better structural preservation. The second graph further confirms that the optimal spectral methods provide stable and high similarity ratios across different network sizes, while the non-optimal approaches experience larger decreases. The third graph compares the distance-based method (d_0) with spectral approaches and shows that the distance-based method can preserve higher similarity in certain cases, especially for larger network configurations. The experiments indicate that spectral-based optimization methods are effective for analyzing and preserving biological network structures. The optimal parameters (σ and σ') provide more stable performance, while distance-based methods may perform better for specific network sizes. These results highlight the importance of selecting an appropriate similarity measure and optimization strategy for accurate analysis of complex biological networks.

Conclusion

This study demonstrates the importance of graph-based approaches in computational biology for analyzing complex biological networks. Using the Laplacian and normalized Laplacian matrices, spectral analysis, eigenvalue sets, protein interaction networks, domain occurrence, edit distance, bipartite matching distance, and spectrum metrics, the structural properties of biological networks can be effectively examined and compared. The analysis of species such as *Homo sapiens* and *Mus musculus* shows that spectral methods provide valuable insights into network organization and similarity. Overall, Graph Machine Learning and graph-theoretic techniques offer powerful tools for understanding biological systems and support applications in protein analysis, disease research, and network reconstruction.

Bibliography

1. Mrinka Zitnik.*Graph Neural Network in Computational Biology*,March 2021
2. Jie Zhou,Ganqu Cui,Shengding Hu,Zhengyan Zhang,Cheng Yang,Zhiyuan Liu,Lifeng Wang,Changcheng Li,Maosong Sun.*Graph neural networks: A review of methods and applications*,2020
3. Jiao Gu.*The spectral Analysis on Biology Networks*, Bled , February 16,2012.