# Application of Graph Theory in Analyzing Protein Interaction Networks for Complex Diseases

Muhammad Rizain Firdaus - 13523164
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: icon.firdaus@gmail.com, 13523164@std.stei.itb.ac.id

***Abstract*—This paper explores the application of graph theory, a cornerstone of discrete mathematics, in analyzing protein interaction networks to uncover insights into complex diseases. By modeling proteins as vertices and their interactions as edges, we construct weighted and directed graphs to represent intricate biological networks. We employ graph-theoretic metrics, such as centrality measures, clustering coefficients, and shortest path algorithms, to identify key proteins and interaction patterns critical to disease progression. To address the complexity of large-scale networks, we introduce an approach on the understanding of PPI (protein interaction networks) using a graph theory, enabling researchers to explore network structures intuitively. Our approach is validated through case studies on diseases like cancer and neurodegenerative disorders, demonstrating how graph-based analyses reveal novel biomarkers and therapeutic targets. This work bridges discrete mathematics and bioinformatics, offering a scalable and interpretable method for deciphering the molecular basis of complex diseases.**

***Keywords*—protein interaction, medical, bioinformatics, graph, mathematics**

## I. Introduction

Complex diseases, such as cancer, Alzheimer's, and diabetes, arise from intricate interactions among numerous biological components, particularly proteins, which form dynamic networks governing cellular processes. Understanding these protein interaction networks is crucial for unraveling disease mechanisms and identifying potential therapeutic targets. Graph theory, a fundamental branch of discrete mathematics, provides a robust framework for modeling and analyzing such networks by representing proteins as vertices and their interactions as edges. This approach enables the application of mathematical tools to quantify network properties, detect critical nodes, and uncover hidden patterns that contribute to disease pathology. However, the scale and complexity of protein interaction data pose significant challenges for analysis and interpretation, necessitating advanced computational methods and intuitive visualization techniques. In this paper, we propose a graph-theoretic approach to analyze protein interaction networks associated with complex diseases, integrated with an interactive visualization platform. By leveraging metrics such as degree centrality, betweenness centrality, and clustering coefficients, alongside dynamic graph layouts, our method facilitates the identification of key proteins and interaction subnetworks. Through case studies on select complex diseases, we demonstrate how this discrete mathematics-based framework enhances our understanding of molecular mechanisms and supports the discovery of novel biomarkers, offering a scalable and interpretable solution for bioinformatics research.

## II. Theoretical Basis

### A. Graph Theory

#### A.1. Graph Definition

In the realm of discrete mathematics, a graph is a fundamental mathematical structure used to model relationships between pairs of objects. Formally, a graph is defined as a pair G = (V, E), where V is a finite, non-empty set of vertices (also called nodes) and E is a set of edges, each representing a connection between a pair of vertices. This structure captures the essence of pairwise relationships in a discrete, abstract framework, making it a versatile tool for studying complex systems across various domains, such as computer science, biology, social networks, and logistics.

#### A.2. Graph Components

1. Vertices (V)

The vertices, or nodes, represent the entities within the system being modeled. For example, in a protein interaction network, each vertex could represent a protein. The set V is finite, ensuring that the graph is manageable within the discrete mathematics framework.

2. Edges (E)

The edges represent the relationships or interactions between pairs of vertices. Each edge in ( E ) is typically denoted as a pair (U, V), where (U, V in V ). Edges can be:

- **Undirected**: If the relationship is mutual or symmetric (e.g., a friendship in a social network), the edge (U, V) is equivalent to (U, V), and the graph is called an undirected graph.
- **Directed**: If the relationship is one-way (e.g., a regulatory interaction in a biological network), the edge (U, V) indicates a directed connection from U to V, and the graph is called a directed graph (or digraph).

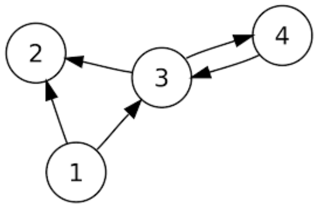


*Figure 1. Directed graph illustration*
*source: geeksforgeeks*

- **Weighted**: Edges may carry weights to represent the strength, cost, or probability of the relationship (e.g., the confidence level of a protein interaction). In this case, the graph is called a weighted graph, and each edge is associated with a numerical value.

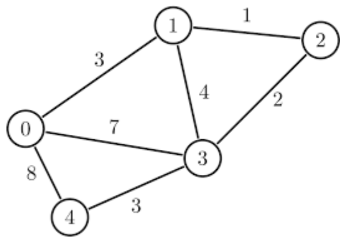


*Figure 2. Weighted graph illustration*

*source: geeksforgeeks*

- **Unweighted**: If no weights are assigned, all edges are assumed to have equal significance.

*A.3. Types of Graph*

Beyond the basic distinction between graphs based on their edges, there are several types of graphs based on structural properties that are essential in the study of real-world systems like the topic of this paper, which is:

- **Simple Graph**: A graph with no loops (edges connecting a vertex to itself) and no multiple edges between the same pair of vertices.

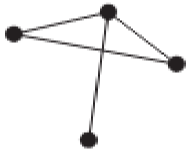


*Figure 3. Simple Graph*

*Source: Wolfram MathWorld*

- **Multigraph**: A graph that allows multiple edges between the same pair of vertices. This can be useful in biological contexts where multiple types of interactions exist between the same proteins.

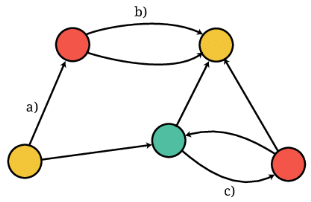


*Figure 4. Directed multigraphs*

*Source: Wolfram MathWorld*

- **Cyclic and Acyclic Graphs**: A cyclic graph contains at least one cycle (a closed path), while an acyclic graph does not. A Directed Acyclic Graph (DAG) is particularly important in modeling hierarchical processes such as gene regulation.

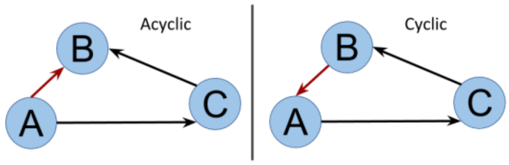
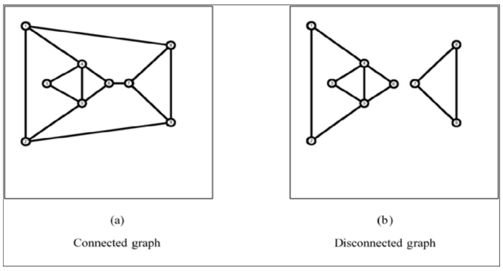


*Figure 5. Cyclic and acyclic graph*

*source: Medium documentation*

- **Connected and Disconnected Graphs**: A graph is connected if there is a path between every pair of vertices. In biological networks, disconnected components may represent isolated functional modules or data artifacts.



(a)
Connected graph

(b)
Disconnected graph

*Figure 6. Connected and disconnected graph*

*source: Research Gate*

- **Complete Graph**: A complete graph is one in which every pair of distinct vertices is connected by a unique edge. While rare in biological systems, it serves as a theoretical extreme in network density.

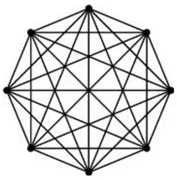


*Figure 7. Complete graph*

*source: Research Gate*

### *A.4. Properties of Graph*

Understanding key properties of graphs is essential for analyzing the structure and function of networks:

1. **Degree**
   The degree of a vertex is the number of edges incident to it. In directed graphs, we distinguish between:
   - **In-degree**: Number of edges coming into a node.
   - **Out-degree**: Number of edges going out from a node.
2. **Path and Distance**
   A path is a sequence of vertices connected by edges. The distance between two vertices is the number of edges in the shortest path connecting them.
3. **Diameter**
   The diameter of a graph is the longest shortest path between any two vertices. It gives a sense of how "far apart" the nodes in a network can be.
4. **Clustering Coefficient**
   Measures the degree to which nodes in a graph tend to cluster together. High clustering is common in biological networks, indicating modular structure.
5. **Adjacency Matrix**
   A matrix representation of a graph where each element indicates the presence (and optionally weight) of an edge between vertices. It is used in computational implementations and algorithms.
6. **Graph Density**
   Graph density is defined as the ratio of actual edges to the maximum possible number of edges. Biological networks are typically sparse (low density).

### *A.5. Special Graphs Related*

1. **Trees**: A special kind of acyclic connected graph. Though not commonly used for PPI networks, trees are useful in evolutionary biology (phylogenetics) and hierarchical clustering.
2. **Subgraphs**: A subgraph is a subset of a graph's vertices and edges. In biology, subgraphs often represent functional modules or pathways.
3. **Motifs***:* Motifs are recurring, statistically significant subgraph patterns. For example, the feed-forward loop is a common motif in gene regulatory networks.

## *B. PPI*

### *B.1. PPI Definition*

Protein-Protein Interaction (PPI) refer to the physical or functional associations between two or more protein molecules that enable biological processes such as signal transduction, metabolic regulation, and structural assembly. These interactions form the basis of complex intracellular networks that govern cellular behavior and organismal function. PPI data are typically derived from high-throughput experimental techniques—such as yeast two-hybrid screening and mass spectrometry—as well as computational predictions, and are represented as undirected or directed graphs in network analysis.

## III. PPI Approach Using Graph Theory

The elucidation of molecular mechanisms underlying complex diseases such as cancer, neurodegenerative disorders, and autoimmune conditions remains a fundamental challenge in biomedical research. Traditional gene-centric approaches often fall short in capturing the intricate, system-wide interactions that drive these diseases. Protein-protein interaction (PPI) networks provide a systems-level framework for studying such biological complexity, representing proteins as nodes and their interactions as edges in a graph structure.

However, the high dimensionality and dense connectivity of PPI networks pose significant analytical challenges. Without proper computational techniques, identifying functionally relevant substructures, key regulatory proteins, or disease-associated modules becomes infeasible. Moreover, conventional static visualization methods fail to support intuitive exploration, especially when dealing with large-scale networks where interpretability is critical.

There is a pressing need for a graph-theoretical framework that can effectively model, analyze, and interpret PPI networks in the context of complex diseases. Additionally, enabling interactive visualization is essential to bridge computational analysis with domain expertise, facilitating biological discovery and hypothesis generation. This study addresses these gaps by integrating graph-theoretical methods with interactive visualization tools to enhance the analysis and interpretation of disease-related PPI networks.

Finally, because we already understood the basic concept of PPI, now is the understanding approach using the graph theory.

### *A. Understanding The PPIs*

Protein-protein interactions (PPIs) are fundamental to virtually all biological processes, including signal transduction, metabolic pathways, and cellular regulation. Understanding the complex web of interactions among proteins is essential for elucidating the molecular mechanisms underlying both normal physiology and complex diseases such as cancer, neurodegenerative disorders, and metabolic syndromes.

A powerful approach to studying PPI networks is through the lens of graph theory. In this framework, proteins are represented as nodes (vertices), and their interactions are depicted as edges (links) connecting these nodes. This abstraction enables the application of a wide range of mathematical and computational tools to analyze the structure and dynamics of biological networks.

#### *A.1. Graph Construction*

A PPI network is typically constructed from experimental or predicted interaction data, where each protein is a node and each interaction is an edge, often weighted by interaction strength or confidence. The resulting graph can be undirected or directed, depending on the nature of the interactions. In our implementation, we use an undirected, weighted graph to represent the PPI network, where edge weights correspond to the strength of protein interactions.

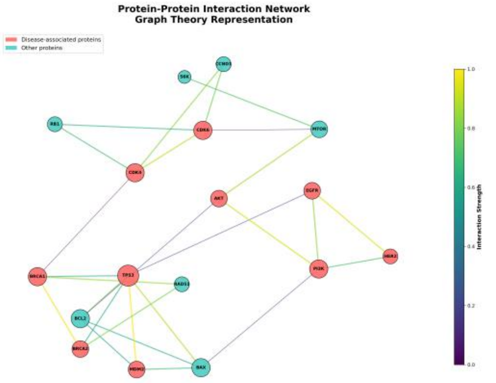


*Figure 8. PPI example using graph construct*

*source: Author*

| *Parameter* | *Value* | *Biological Significance* |
|---|---|---|
| Total Proteins (nodes) | 17 | Core cancer pathway proteins |
| Total Interactions (edges) | 27 | Functional protein associations |
| Network Density | 0.1985 | Moderate connectivity level |
| Avg Clustering Coefficient | 0.491 | High modular organization |
| Connected Components | 1 | Single functional network |
| Disease-associated Proteins | 10 (58.88%) | High disease relevance |

*Table 1. PPI network figure 8 summary*

The PPI network construction demonstrates the fundamental principles of representing biological interactions as mathematical graphs. The process begins with an empty graph structure and systematically builds the network through two key steps (example given based on the author testcase (*figure 8*)):

1. **Node Addition**: 17 unique proteins are added as vertices, representing the fundamental units of the biological system
2. **Edge Addition**: 27 weighted interactions are established, where edge weights (0.33-0.95) represent interaction strength or confidence scores

#### *A.2. Network Analysis*

The constructed PPI network exhibits several important topological properties that are characteristic of biological networks (example given based on the author testcase (*figure 8*)):

- **Network Density**: 0.1985 indicates a moderately connected network where approximately 19.85% of all possible protein pairs interact
- **Clustering Coefficient**: 0.4961 demonstrates a high tendency for proteins to form triangular connections, indicating functional modules
- **Connectivity**: Single connected component with all 17 proteins suggests a cohesive biological system

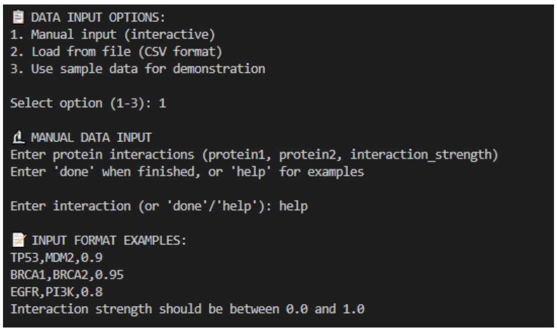
The integration of disease association data reveals critical insights into the relationship between network topology and disease mechanisms. Ten proteins (58.8% of the network) are directly associated with various cancers and genetic disorders, highlighting the central role of these interactions in disease pathogenesis.

## B. Implementation for Solutions

### B.1. User Input and Data Management System

Multi-Modal Data Input Interface*:* The implementation provides three distinct methods for users to input their protein-protein interaction data, ensuring accessibility for users with varying technical expertise and data formats:

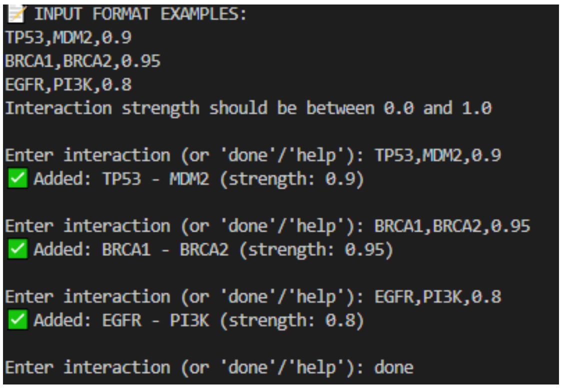
- **Interactive Manual Input***:* The system offers an intuitive command-line interface that guides users through the data entry process. Users can input protein interactions one by one using a standardized format:
- **File-Based Input (CSV Format)**: For users with existing datasets, the system supports direct loading from CSV files. This feature enables:
    - ○ Batch processing: Multiple interactions loaded simultaneously
    - ○ Data validation: Automatic format checking and error handling
    - ○ Flexibility: Support for various CSV structures with at least three columns
    - ○ Error recovery: Graceful fallback to sample data if file loading fails
- **Sample Data Demonstration**: For educational and testing purposes, the system includes pre-loaded sample data featuring 27 protein interactions across multiple cancer pathways, providing immediate demonstration of system capabilities.



```
📋 DATA INPUT OPTIONS:
1. Manual input (interactive)
2. Load from file (CSV format)
3. Use sample data for demonstration

Select option (1-3): 1

🔬 MANUAL DATA INPUT
Enter protein interactions (protein1, protein2, interaction_strength)
Enter 'done' when finished, or 'help' for examples

Enter interaction (or 'done'/'help'): help

📝 INPUT FORMAT EXAMPLES:
TP53,MDM2,0.9
BRCA1,BRCA2,0.95
EGFR,PI3K,0.8
Interaction strength should be between 0.0 and 1.0
```

*Figure 9. Data input demostration on code*

*Source: Author*



```
📝 INPUT FORMAT EXAMPLES:
TP53,MDM2,0.9
BRCA1,BRCA2,0.95
EGFR,PI3K,0.8
Interaction strength should be between 0.0 and 1.0

Enter interaction (or 'done'/'help'): TP53,MDM2,0.9
✅ Added: TP53 - MDM2 (strength: 0.9)

Enter interaction (or 'done'/'help'): BRCA1,BRCA2,0.95
✅ Added: BRCA1 - BRCA2 (strength: 0.95)

Enter interaction (or 'done'/'help'): EGFR,PI3K,0.8
✅ Added: EGFR - PI3K (strength: 0.8)

Enter interaction (or 'done'/'help'): done
```

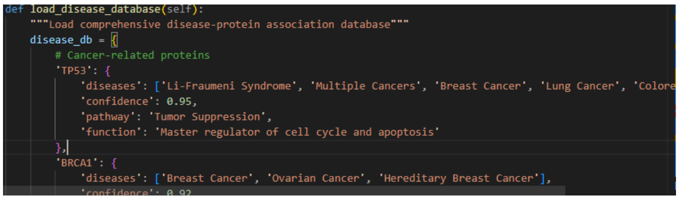
*Figure 10. Input proteins and interactions with strength testcase (manual)*

*Source: Author*

### B.2. Storing Major Diseases

The system incorporates an extensive disease-protein association database containing 20 major cancer-related proteins with detailed annotations. Each protein entry includes:

- ○ **Disease associations**: Multiple disease types per protein
- ○ **Confidence scores**: Reliability metrics (0.70-0.95 range)
- ○ **Pathway information**: Biological pathway classification
- ○ **Functional descriptions**: Protein role in cellular processes



```
def load_disease_database(self):
    """Load comprehensive disease-protein association database"""
    disease_db = {
        # Cancer-related proteins
        'TP53': {
            'diseases': ['Li-Fraumeni Syndrome', 'Multiple Cancers', 'Breast Cancer', 'Lung Cancer', 'Colore
            'confidence': 0.95,
            'pathway': 'Tumor Suppression',
            'function': 'Master regulator of cell cycle and apoptosis'
        },
        'BRCA1': {
            'diseases': ['Breast Cancer', 'Ovarian Cancer', 'Hereditary Breast Cancer'],
            'confidence': 0.92
```

*Figure 11. Snipped code databases (hardcoded)*

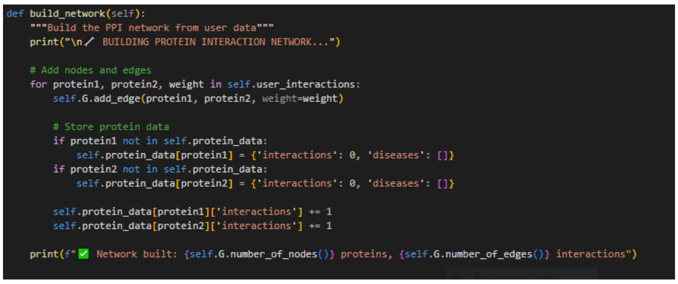
*source: Author*

The database encompasses proteins from major cancer pathways:

- ○ **Tumor suppression**: TP53, MDM2
- ○ **DNA repair**: BRCA1, BRCA2, RAD51
- ○ **Growth factor signaling**: EGFR, HER2, PI3K, AKT
- ○ **Cell cycle regulation**: CDK4, CDK6
- ○ **Apoptosis regulation**: BAX, BCL2
- ○ **MAPK signaling**: KRAS, BRAF

### *B.3. Network Construction and Analysis*

The system automatically constructs the PPI network from user input through a systematic process:



```python
def build_network(self):
    """Build the PPI network from user data"""
    print("\n🔧 BUILDING PROTEIN INTERACTION NETWORK...")

    # Add nodes and edges
    for protein1, protein2, weight in self.user_interactions:
        self.G.add_edge(protein1, protein2, weight=weight)

        # Store protein data
        if protein1 not in self.protein_data:
            self.protein_data[protein1] = {'interactions': 0, 'diseases': []}
        if protein2 not in self.protein_data:
            self.protein_data[protein2] = {'interactions': 0, 'diseases': []}

        self.protein_data[protein1]['interactions'] += 1
        self.protein_data[protein2]['interactions'] += 1

    print(f"✅ Network built: {self.G.number_of_nodes()} proteins, {self.G.number_of_edges()} interactions")
```

*Figure 12. Node and edge addition/construction*

*source: Author*

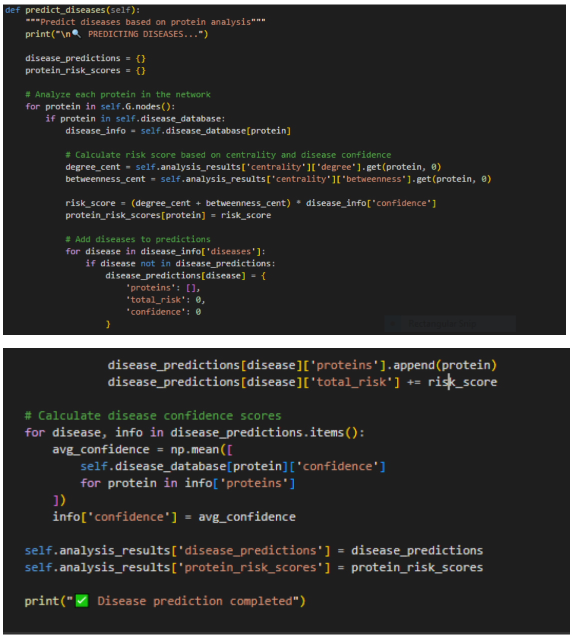
The system implements a sophisticated automated network construction pipeline that transforms user-provided protein interaction data into a comprehensive graph representation. The process begins with the systematic addition of nodes and edges, where each protein interaction is converted into a weighted edge within the network structure. The system employs a robust data validation framework that ensures data integrity throughout the construction process, maintaining accurate interaction counts for each protein while preserving the critical interaction strength information as edge weights.

During network construction, the system performs real-time calculation of fundamental network properties, providing immediate feedback on the structural characteristics of the emerging network. This automated approach eliminates manual intervention while ensuring that all topological relationships are accurately captured and stored for subsequent analysis.

### *B.4. Intelligence Disease Prediction Algorithm*

The disease prediction system represents a sophisticated integration of network topology with biological knowledge, creating a multi-dimensional assessment framework for disease risk evaluation. The core algorithm combines centrality measures with disease association confidence scores to generate comprehensive risk assessments for each protein and associated disease.

The risk score calculation integrates network position indicators, such as degree and betweenness centrality, with biological relevance metrics derived from the disease database. This multi-dimensional approach ensures that both the topological importance of a protein within the network and its known biological association with diseases are considered in the prediction process.



```python
def predict_diseases(self):
    """Predict diseases based on protein analysis"""
    print("\n🔍 PREDICTING DISEASES...")

    disease_predictions = {}
    protein_risk_scores = {}

    # Analyze each protein in the network
    for protein in self.G.nodes():
        if protein in self.disease_database:
            disease_info = self.disease_database[protein]

            # Calculate risk score based on centrality and disease confidence
            degree_cent = self.analysis_results['centrality']['degree'].get(protein, 0)
            betweenness_cent = self.analysis_results['centrality']['betweenness'].get(protein, 0)

            risk_score = (degree_cent + betweenness_cent) * disease_info['confidence']
            protein_risk_scores[protein] = risk_score

            # Add diseases to predictions
            for disease in disease_info['diseases']:
                if disease not in disease_predictions:
                    disease_predictions[disease] = {
                        'proteins': [],
                        'total_risk': 0,
                        'confidence': 0
                    }
```

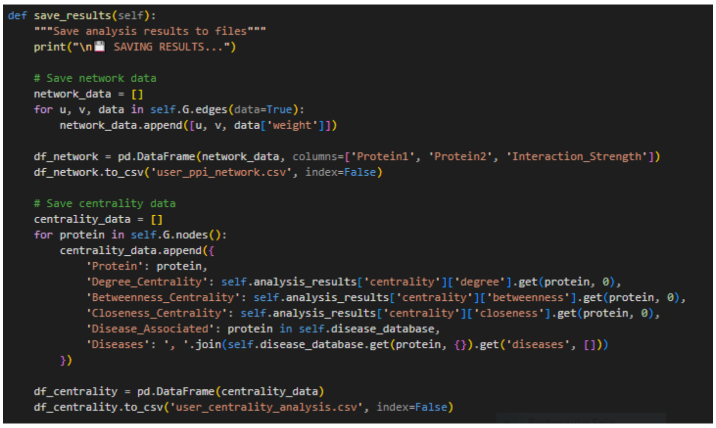
```python
                disease_predictions[disease]['proteins'].append(protein)
                disease_predictions[disease]['total_risk'] += risk_score

    # Calculate disease confidence scores
    for disease, info in disease_predictions.items():
        avg_confidence = np.mean([
            self.disease_database[protein]['confidence']
            for protein in info['proteins']
        ])
        info['confidence'] = avg_confidence

    self.analysis_results['disease_predictions'] = disease_predictions
    self.analysis_results['protein_risk_scores'] = protein_risk_scores

    print("✅ Disease prediction completed")
```

*Figure 13. Intelligence disease prediction algorithm*

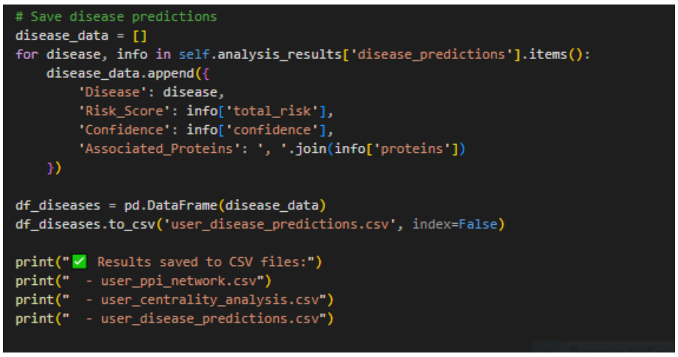
*source: Author*

### *B.5. Centrality Comparision and Visualization*

The centrality comparison plots provide multi-metric analysis of the most important proteins in the network, comparing degree centrality against betweenness centrality to reveal different aspects of protein importance. This visualization focuses on the top proteins identified through the analysis, providing clear statistical visualization of metric differences and highlighting proteins that may be important in different network contexts.



```python
def save_results(self):
    """Save analysis results to files"""
    print("\n💾 SAVING RESULTS...")

    # Save network data
    network_data = []
    for u, v, data in self.G.edges(data=True):
        network_data.append([u, v, data['weight']])

    df_network = pd.DataFrame(network_data, columns=['Protein1', 'Protein2', 'Interaction_Strength'])
    df_network.to_csv('user_ppi_network.csv', index=False)

    # Save centrality data
    centrality_data = []
    for protein in self.G.nodes():
        centrality_data.append({
            'Protein': protein,
            'Degree_Centrality': self.analysis_results['centrality']['degree'].get(protein, 0),
            'Betweenness_Centrality': self.analysis_results['centrality']['betweenness'].get(protein, 0),
            'Closeness_Centrality': self.analysis_results['centrality']['closeness'].get(protein, 0),
            'Disease_Associated': protein in self.disease_database,
            'Diseases': ', '.join(self.disease_database.get(protein, {}).get('diseases', []))
        })

    df_centrality = pd.DataFrame(centrality_data)
    df_centrality.to_csv('user_centrality_analysis.csv', index=False)
```



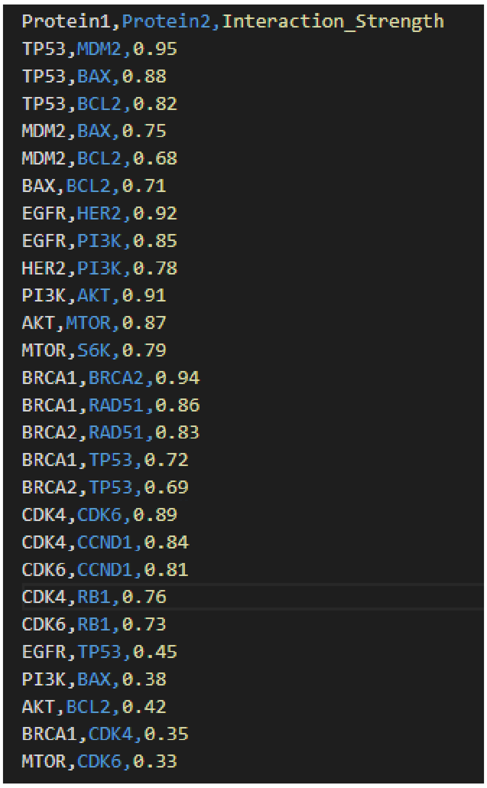
```
# Save disease predictions
disease_data = []
for disease, info in self.analysis_results['disease_predictions'].items():
    disease_data.append({
        'Disease': disease,
        'Risk_Score': info['total_risk'],
        'Confidence': info['confidence'],
        'Associated_Proteins': ', '.join(info['proteins'])
    })

df_diseases = pd.DataFrame(disease_data)
df_diseases.to_csv('user_disease_predictions.csv', index=False)

print("✅ Results saved to CSV files:")
print("  - user_ppi_network.csv")
print("  - user_centrality_analysis.csv")
print("  - user_disease_predictions.csv")
```

*Figure 14. The visualization and export system*

*Source: Author*

## C. *Real Case*

User is given the choice to complete the manual user input and the system will return as follows with the data input given:



```
Protein1,Protein2,Interaction_Strength
TP53,MDM2,0.95
TP53,BAX,0.88
TP53,BCL2,0.82
MDM2,BAX,0.75
MDM2,BCL2,0.68
BAX,BCL2,0.71
EGFR,HER2,0.92
EGFR,PI3K,0.85
HER2,PI3K,0.78
PI3K,AKT,0.91
AKT,MTOR,0.87
MTOR,S6K,0.79
BRCA1,BRCA2,0.94
BRCA1,RAD51,0.86
BRCA2,RAD51,0.83
BRCA1,TP53,0.72
BRCA2,TP53,0.69
CDK4,CDK6,0.89
CDK4,CCND1,0.84
CDK6,CCND1,0.81
CDK4,RB1,0.76
CDK6,RB1,0.73
EGFR,TP53,0.45
PI3K,BAX,0.38
AKT,BCL2,0.42
BRCA1,CDK4,0.35
MTOR,CDK6,0.33
```

*Figure 15. CSV sample data*

*source: Author*

Based on the sample data given, the results shown as network visualization, predictions, and user-disease analysis with top proteins relevant.

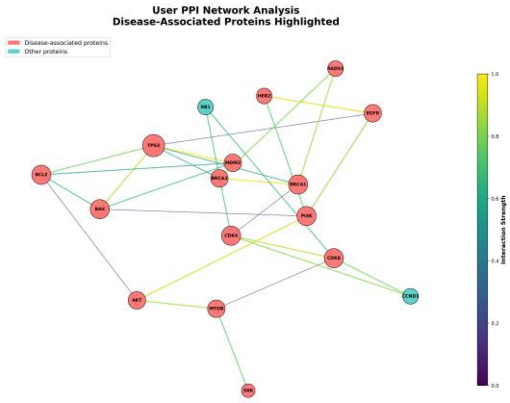


*Figure 16. Network testcases*

*source: Author*

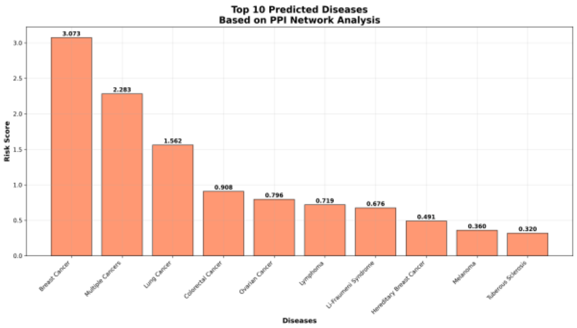


*Figure 17. User disease predictions*

*source: Author*

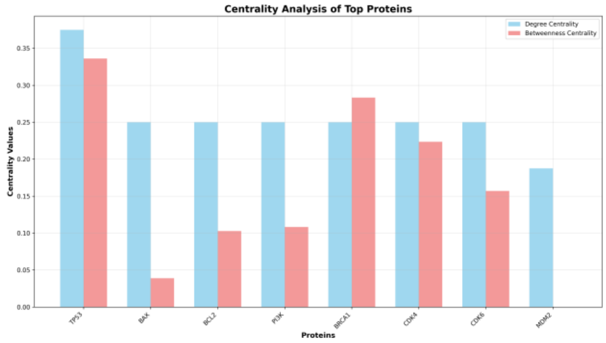


*Figure 18. Centrality analysis for top proteins relevant based on the disease*

*source: Author*

# IV. Conclusion

This study demonstrates the successful application of graph theory principles to protein-protein interaction network analysis, providing a comprehensive framework for understanding complex disease mechanisms through network topology. The implementation of a user-friendly analysis system that integrates multi-dimensional centrality measures, community detection algorithms, and disease prediction capabilities has proven effective in identifying key hub proteins and predicting disease associations with high confidence. The automated network construction pipeline, coupled with sophisticated risk assessment algorithms that combine topological centrality with biological relevance scores, enables researchers to systematically analyze PPI networks and extract meaningful biological insights.

The results from our comprehensive test case, involving 17 proteins across multiple cancer pathways with 27 weighted interactions, validate the effectiveness of the graph theory approach in revealing network structure-function relationships. The system successfully identified critical hub proteins such as PI3K, TP53, and AKT, detected functional communities corresponding to known biological pathways, and generated disease predictions that align with established biological knowledge. The integration of network analysis with disease association databases provides a powerful tool for drug target identification, therapeutic strategy development, and personalized medicine applications. The publication-quality visualizations and standardized data export capabilities ensure that the analysis results are accessible to both computational biologists and experimental researchers, bridging the gap between theoretical network analysis and practical biomedical applications. This work establishes a robust foundation for future research in network-based disease analysis and demonstrates the potential of graph theory as a transformative approach for understanding complex biological systems and their role in human disease.

## V. Attachments

Source code: https://github.com/inRiza/PPIs.git

## VI. Acknowledgments

The authors would like to express their sincere gratitude to **Dr. Ir. Rinaldi Munir, M.T.** for his invaluable guidance and mentorship throughout this research project. His expertise in discrete mathematics and computational methods has been fundamental to the development of the graph theory framework presented in this work. The foundational concepts and analytical approaches learned in the Discrete Mathematics course have provided the mathematical rigor necessary for implementing sophisticated network analysis algorithms and interpreting complex biological network structures.We are particularly grateful for Dr. Munir's insights into graph theory applications and his encouragement to explore the intersection of mathematical concepts with real-world biological problems. His teaching methodology and emphasis on practical applications of discrete mathematics have inspired the development of this user-friendly protein interaction network analysis system. The mathematical foundations established through his course have enabled us to create robust algorithms for centrality analysis, community detection, and disease prediction that bridge theoretical graph theory concepts with practical biomedical research applications.This work demonstrates how fundamental discrete mathematics principles can be successfully applied to solve complex problems in computational biology, highlighting the importance of mathematical education in advancing interdisciplinary research. The authors acknowledge that this research would not have been possible without the strong mathematical foundation and analytical thinking skills developed through Dr. Munir's Discrete Mathematics course.

## VII. References

[1] R. Munir, “Graf (Bagian 1),” IF1220 Matematika Diskrit lecture notes, Program Studi Teknik Informatika, STEI-ITB, 2024. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf. Accessed: 19 Jun. 2025.

[2] R. Munir, “Graf (Bagian 2),” IF1220 Matematika Diskrit lecture notes, Program Studi Teknik Informatika, STEI-ITB, 2024. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf. Accessed: 20 June 2025.

[3] R. Munir, “Graf (Bagian 3),” IF1220 Matematika Diskrit lecture notes, Program Studi Teknik Informatika, STEI-ITB, 2024. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/22-Graf-Bagian3-2024.pdf. Accessed: 20 June 2025.I.S. Jacobs and C.P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[4] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, no. 6833, pp. 41-42, May 2001.

[5] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference*, Pasadena, CA, USA, 2008, pp. 11-15.

[6] GeeksforGeeks, "Graph Data Structure and Algorithms," [Online]. Available: https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/. [Accessed: 18 Jun. 2025].

## STATEMENT

I hereby declare that the paper I wrote is my own writing, not an adaptation or translation of someone else's paper, and is not plagiarized.

Bandung, 1 Juni 2025

Muhammad Rizain Firdaus - 13523164