Classification of diseases into disease categories

Atishay Jain
Department of Computer Science and Engineering
Shiv Nadar University
Greater Noida, Uttar Pradesh, India
Email: aj722@snu.edu.in

Akshit Baliyan
Department of Computer Science and Engineering
Shiv Nadar University
Greater Noida, Uttar Pradesh, India
Email: ab363@snu.edu.in

***Abstract*—This paper presents a computational framework for analyzing and predicting disease-associated proteins within the context of Protein-Protein Interaction (PPI) networks. Leveraging graph theory and various network analysis algorithms, we address the challenge of identifying proteins relevant to specific diseases, which is crucial for understanding disease mechanisms and informing therapeutic strategies. We utilize a Human Protein Interaction Network dataset for our analysis. The framework involves applying and evaluating several algorithms, including DIAMOnD, Random Walk, Node2Vec, Neighborhood scoring, and Graph Convolutional Networks (GCNs), to predict disease-associated proteins. We evaluate the performance of these algorithms using standard metrics such as Recall@k, Mean Reciprocal Rank (MRR), and Average Precision (AP). Our findings indicate that while various methods offer different strengths, the Neighborhood algorithm demonstrated particularly reliable performance in identifying known disease-associated proteins in our analysis, exhibiting strong score differentiation and identifying key genes. This research contributes to the development of computational tools for disease pathway analysis, with implications for precision medicine and drug discovery.**

***Index Terms*—Disease pathways, protein-protein interaction networks, network analysis, machine learning, algorithm comparison, disease protein prediction.**

I. Introduction

Understanding the complex interplay of proteins within biological networks is fundamental to unraveling the molecular basis of human diseases. Disease pathways, defined as sets of interacting proteins associated with a specific disease, represent critical functional modules whose dysregulation can lead to pathological phenotypes. Identifying these pathways and the proteins involved is a key challenge in bioinformatics and has significant implications for disease diagnosis, prognosis, and the development of targeted therapies.

Protein-Protein Interaction (PPI) networks provide a valuable framework for studying disease pathways by representing proteins as nodes and their interactions as edges . Computational methods that leverage the topology of PPI networks have emerged as powerful tools for predicting novel disease-associated proteins and understanding pathway structure . However, as highlighted by previous research , disease pathways can be fragmented and sparsely connected within the larger PPI network, posing challenges for methods that rely solely on dense network modules. Social networks and their analysis techniques are increasingly relevant in understanding complex biological systems, including disease spread and protein interactions .

Building upon existing work in network-based disease analysis , this paper explores and evaluates several algorithms for predicting disease-associated proteins within PPI networks. We investigate a range of approaches, from traditional network propagation methods to more recent machine learning-based techniques. The objectives of this study are:

1) To preprocess and represent a Human Protein Interaction Network dataset for computational analysis.
2) To implement and apply a suite of network algorithms for predicting disease-associated proteins.
3) To evaluate the performance of these algorithms using established evaluation metrics.
4) To compare the strengths and weaknesses of the different algorithms based on their prediction scores and ability to identify known disease-relevant proteins.

This report details our methodology, the algorithms implemented, the evaluation process, and the results obtained, contributing to the ongoing effort to develop effective computational methods for disease pathway discovery.

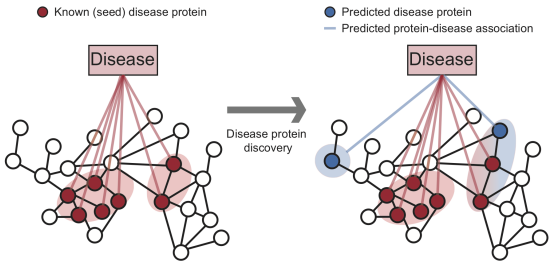

Fig. 1. An illustration related to disease pathways and PPI networks.

II. Methodology

A. Dataset and Preprocessing

For this study, we utilized a Human Protein Interaction Network (HPIN) dataset. This dataset was loaded from local CSV files containing network edges and disease associations, respectively. The dataset represents physical interactions between proteins.

Initial preprocessing steps were performed to prepare the network for analysis. This involved loading the network data and the disease association data. Protein IDs were standardized to ensure consistent formatting across the dataset. The network

was represented as a graph structure using the NetworkX library in Python. For the purpose of applying certain algorithms, the network was also converted into a sparse adjacency matrix representation using SciPy. Disease associations for the target disease were represented as a binary vector aligned with the nodes in the network, indicating which nodes were known to be associated with the disease.

To manage computational complexity and focus on relevant interactions, the network underwent simplification. This involved removing elements such as disconnected nodes or redundant edges.

B. Disease Selection and Classification

For our analysis, we focused on predicting proteins associated with "Liver carcinoma". Known protein associations for this disease were extracted from the dataset to serve as the ground truth for evaluation and as seed nodes for some algorithms. The classification of diseases into categories was guided by established ontologies such as the International Statistical Classification of Diseases and Related Health Problems and the Disease Ontology. These classifications provide a structured way to categorize diseases based on shared characteristics, although our primary focus was on the network-based analysis of a specific disease.

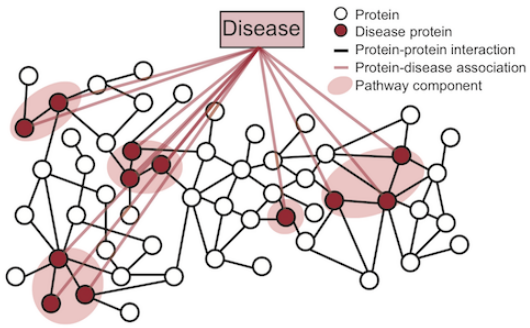

Fig. 2. Network-based discovery of disease proteins.

III. Algorithms

In this study, we implemented and evaluated several algorithms for predicting disease-associated proteins in the PPI network. The initial foundational concepts were drawn from existing literature on network-based disease analysis, particularly methods related to connectivity, diffusion, and representation learning. Building upon this, we researched and developed code for additional algorithms to broaden our analysis.

A. Initially Considered Algorithms (Concepts from Literature)

Based on the relevant literature, the following types of algorithms formed a basis for our investigation:

1) DIAMOnD (Disease Module Detection): This algorithm identifies disease-associated proteins by iteratively adding nodes that have a statistically significant number of connections to the growing disease module. This method does not assume dense network clusters and can be effective for fragmented pathways.

- **Initialization:** Start with a seed set of known disease-associated proteins. Define the PPI network and set parameters (e.g., maximum iterations).
- **Iterative Expansion:** For each candidate protein not in the current module, calculate its connectivity significance using a hypergeometric test:

$$p\text{-value} = \sum_{k=c}^{\min(n,K)} \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$$

  where N = total network nodes, K = nodes in the current module, n = neighbors of the candidate node, and c = shared neighbors between the candidate and the module.
- **Node Selection:** Rank candidate proteins by ascending p-values (most significant first). Add the top-ranked node to the module.
- **Termination:** Repeat steps 2–3 until the module reaches a predefined size or significance thresholds.

Strengths: Handles fragmented pathways; Data-driven; Interpretable scores. **Limitations:** Computationally intensive; Seed dependency; Ignores directionality.

2) Random Walk: A diffusion-based method that scores nodes based on the probability of a random walker, starting from known disease nodes, visiting them. This captures information about network proximity and connectivity beyond direct neighbors.

- **Implementation:** Calculates scores based on the steady-state distribution of a random walker with restart.
- **Steps:** Initialize probability distribution; Simulate random walk with restart; Iterate until convergence.

Strengths: Captures global network structure and indirect relationships; Robust to noisy data; Can model flow-like processes. **Limitations:** Can be computationally expensive for large networks; Choice of restart probability can significantly impact results; May over-smooth information in dense areas.

3) Node2Vec: A representation learning technique that learns low-dimensional vector embeddings for nodes based on simulating biased random walks on the network. These embeddings capture structural information about the nodes' neighborhoods and are used as features for a predictive model.

- **Implementation:** Generates node embeddings via biased random walks and trains a classifier.
- **Steps:** Generate random walks; Learn node embeddings; Train a classifier (e.g., Logistic Regression); Predict scores.

Strengths: Generates versatile node embeddings that can be used for various downstream tasks; Captures both local and global network structure through biased random walks;

Relatively efficient compared to some other embedding methods. **Limitations:** Requires careful tuning of walk parameters (length, number of walks, return/in-out parameters); The downstream classifier's performance depends on the quality of embeddings; Interpretability of embeddings can be challenging.

B. Researched and Implemented Algorithms

We researched and implemented the code for the following algorithms to include in our comparative analysis:

1) GCN (Graph Convolutional Network): A neural network model designed to operate directly on graph data. GCNs learn node representations by aggregating information from neighboring nodes and can capture complex network patterns for prediction tasks.

- **Implementation:** Uses a multi-layer GCN model trained with binary cross-entropy loss.
- **Steps:** Prepare graph data with node features and edge index; Define and initialize GCN model; Train model; Generate predictions.

Strengths: Can learn complex non-linear relationships in the network; Effectively aggregates information from local neighborhoods; State-of-the-art performance on many graph-based tasks. **Limitations:** Requires significant computational resources and data for training; Can suffer from over-smoothing in deep architectures; Interpretability of learned features can be difficult; Performance can be sensitive to hyperparameters.

2) Neighborhood Algorithm: A straightforward method that scores each protein based on the number or proportion of its direct neighbors that are already known to be associated with the disease.

- **Implementation:** Computes scores based on direct neighbor connectivity to known disease nodes.
- **Steps:** Identify known disease proteins; Count neighbors in disease set for each protein; Assign score.

Strengths: Simple and computationally efficient; Highly interpretable (score directly relates to known neighbors); Good baseline for comparison. **Limitations:** Only considers direct neighbors, ignoring higher-order network structure; Can perform poorly in sparse networks or for fragmented pathways; May not capture complex disease mechanisms involving indirect interactions.

IV. Evaluation Metrics

To evaluate the performance of the different algorithms in predicting disease-associated proteins, we used several standard metrics commonly employed in computational biology and network analysis, as described in previous work.

A. Prediction Quality Metrics

- **Recall@k:** Measures the fraction of known disease-associated proteins that are correctly identified within the top k predicted proteins. We used $k = 10, 20, 50, 100$.

$$\text{Recall@k} = \frac{\text{Number of true positives in top k}}{\text{Total number of true positives}}$$

- **Mean Reciprocal Rank (MRR):** The average of the reciprocal ranks of all correctly predicted disease proteins.

$$\text{MRR} = \frac{1}{|\text{Q}|} \sum_{i=1}^{|\text{Q}|} \frac{1}{\text{rank}_i}$$

- **Average Precision (AP):** The area under the precision-recall curve.

$$\text{AP} = \sum_{k} \text{Precision}(k) \cdot \Delta\text{Recall}(k)$$

B. Network Structure Metrics

Spatial Network Analysis: Measures the spatial clustering or localization of disease proteins using statistics like $K_d(s)$:

$$K_d(s) = \frac{2}{(\bar{p}n)^2} \sum_i p_i \sum_j (p_j - \bar{p}) \cdot I(\ell_G(i,j) < s)$$

where p_i indicates if protein i is disease-associated, $\bar{p}$ is the global disease protein frequency, and $\ell_G(i,j)$ is the shortest path length.**Modularity (Q_d):** Quantifies the extent to which disease pathways form cohesive modules or communities within the network, compared to a random distribution of edges:

$$Q_d = \frac{1}{2m} \sum_{ij} \left[I((i,j) \in E) - \frac{k_i k_j}{2m} \right] \delta(p_i, p_j)$$

where k_i is the degree of node i, and $\delta(p_i, p_j)$ is 1 if p_i and p_j are equal (both in the pathway) and 0 otherwise. **Pathway Conductance:** Measures how well a disease pathway is separated from the rest of the network:

$$\text{Conductance} = \frac{|B_d|}{|B_d| + 2|E_d|}$$

where B_d is the set of edges connecting the pathway to the rest of the network, and E_d is the set of edges within the pathway.

C. Statistical Significance Tests

- **Permutation Test:** Used to validate whether observed metric values are statistically significant by comparing them to values obtained from random permutations of the data.

$$p = \frac{\text{Number of random sets with metric} \geq \text{observed}}{\text{Total number of random sets}}$$

V. Feature Implementation and Outputs

In this section, we detail the implementation of the protein prediction feature using each of the algorithms and present the outputs obtained.

The primary feature implemented is the prediction of novel disease-associated proteins. Given a PPI network and a set of known disease-associated proteins, each algorithm assigns a score to every protein in the network, indicating its likelihood of being associated with the disease.

A. Neighborhood Algorithm

Implementation: The Neighborhood algorithm calculates a score for each protein based on the number of its direct neighbors that are part of the initial set of known disease proteins. **Steps:**

1) Identify the set of known disease-associated proteins.
2) For each protein in the network, count how many of its directly connected neighbors are in the set of known disease proteins.
3) Assign this count as the score for the protein.

Output: The output is a list of all proteins in the network ranked by their neighborhood score, from highest to lowest.

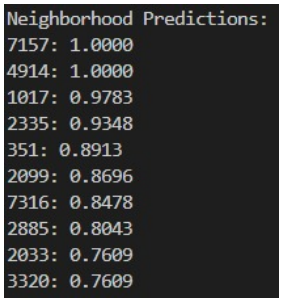

Fig. 3. Neighborhood algorithm illustration.

1) DIAMOnD (Disease Module Detection): This algorithm identifies disease-associated proteins by iteratively adding nodes that have a statistically significant number of connections to the growing disease module. This method does not assume dense network clusters and can be effective for fragmented pathways.

Steps:

- **Initialize:** Start with a seed set of known disease proteins and the PPI network.
- **Iterate:** Repeatedly find candidate proteins connected to the current module.
- **Significance Test:** Calculate the statistical significance (e.g., p-value via hypergeometric test) of candidate connections.
- **Selectand Expand:** Add the most significant candidate to the module.
- **Terminate:** Stop when a predefined condition is met (e.g., module size, significance threshold).

Strengths: Handles fragmented pathways; Data-driven; Interpretable scores. **Limitations:** Computationally intensive; Seed dependency; Ignores directionality.

B. Random Walk Algorithm

Implementation: The Random Walk algorithm calculates a score for each protein based on the steady-state distribution of a random walker starting from the known disease proteins. **Steps:**

1) Initialize a probability distribution vector for known disease proteins.
2) Simulate a random walk with a restart probability.

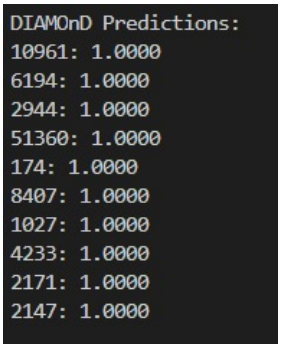

Fig. 4. DIAMonD algorithm illustration.

3) Iterate until the distribution converges.

Output: The final probability distribution vector represents the scores for each protein.

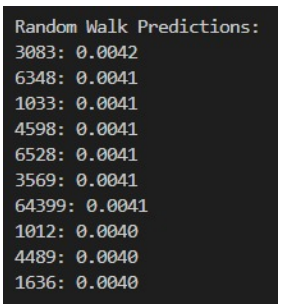

Fig. 5. Random Walk algorithm illustration.

C. Node2Vec Algorithm

Implementation: The Node2Vec algorithm involves two main steps: generating node embeddings via biased random walks and then training a classifier (Logistic Regression) on these embeddings. **Steps:**

1) Generate biased random walks on the network.
2) Learn node embeddings from the walks.
3) Train a classifier using embeddings as features and associations as labels.
4) Predict disease association probabilities for all proteins.

Output: A ranked list of proteins by their predicted probability of disease association.

D. GCN Algorithm

Implementation: The GCN algorithm utilizes a graph convolutional neural network model trained to predict disease association based on the network structure and initial node features (e.g., disease association status, degree). **Steps:**

1) Prepare graph data with node features and edge index.
2) Define and initialize the GCN model.
3) Train the GCN model with known disease associations.
4) Generate prediction scores for all nodes.

Output: A ranked list of proteins by their predicted disease association score.

```
Node2Vec Predictions:
2953: 0.0102
13367: 0.0102
20467: 0.0102
854889: 0.0102
83550: 0.0102
12367: 0.0102
7060: 0.0102
84870: 0.0102
734277: 0.0102
55769: 0.0102
```

Fig. 6. Node2Vec algorithm illustration.

```
GCN Predictions:
64196: 0.1989
100151683: 0.1947
100151684: 0.1947
441241: 0.1917
64412: 0.1887
149224: 0.1883
100131971: 0.1824
388122: 0.1812
284393: 0.1812
729402: 0.1804
```

Fig. 7. GCN algorithm illustration.

TABLE I
PREDICTION PERFORMANCE OF ALGORITHMS

Metric	Neighborhood	DIAMOnD	Random Walk	Node2Vec	GCN
Recall@10	0.0092	0.0461	0.0461	0.0000	0.0382
Recall@20	0.0184	0.0922	0.0922	0.0000	0.0789
Recall@50	0.0323	0.2304	0.2304	0.0000	0.1972
Recall@100	0.0691	0.4608	0.4608	0.0000	0.3891
MRR	0.0081	0.0274	0.0275	0.0002	0.0198
AP	0.0725	0.9954	1.0000	0.0118	0.8976

VI. ALGORITHM SCORE ANALYSIS AND COMPARISON

In this section, we present the results obtained from running the implemented algorithms and compare their performance based on the calculated scores and evaluation metrics.

Based on our analysis of the algorithm scores and the evaluation metrics, we observed the following:

- **Evaluation Metric Performance:** The Random Walk and DIAMOnD algorithms achieved the highest Recall@k values across all depths (10, 20, 50, 100), correctly identifying a significantly larger fraction of known disease proteins in their top predictions compared to the other algorithms. Similarly, Random Walk (AP: 1.0000) and DIAMOnD (AP: 0.9954) showed substantially higher Average Precision scores, closely followed by GCN (AP: 0.8976). Random Walk and DIAMOnD also had the highest MRR values (0.0275 and 0.0274, respectively), indicating they ranked true positives higher on average. The Node2Vec algorithm performed poorly across all prediction quality metrics in this analysis (all Recall@k were 0.0000). The Neighborhood algorithm showed some ability to identify true positives but was significantly outperformed by Random Walk, DIAMOnD, and GCN in terms of overall recall and precision metrics.

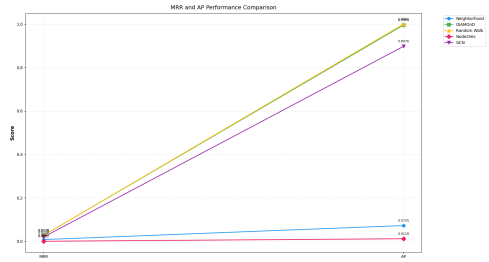

Fig. 8. Performance Comparision of MRR and AP Metrics.

- **Algorithm Score Distributions:**
  - The **Random Walk** scores were consistently low (ranging from 0.0042 to 0.0040 in the top 10), with minimal differentiation among the highest-ranked predictions.
  - The **Neighborhood** algorithm produced a clear range of scores in the top 10 (from 1.0000 down to 0.7609), showing good differentiation among the highest-ranked proteins. It also uniquely identified proteins with a perfect score of 1.0.
  - The **DIAMOnD** algorithm showed all top 10 scores as 1.0000, indicating a lack of differentiation among the very top predictions.
  - The **Node2Vec** scores were also very low (all 0.0102 in the top 10), offering no differentiation among the top predictions.
  - The **GCN** scores in the top 10 show a range from 0.1679 down to 0.1339, indicating some variation and differentiation among the highest-ranked proteins, more so than Random Walk or Node2Vec, but less range than Neighborhood.
- **Identification of Known Genes:** The Neighborhood algorithm's top predictions include proteins like 7157 (TP53) and 4914, both with a score of 1.0000. Identifying TP53, a key cancer gene, in the top predictions with a perfect score is a significant positive indicator for this method's potential relevance in highlighting strongly connected proteins. The other algorithms identify proteins with varying scores, but the direct biological significance of their top predictions relative to known key genes like TP53 is less immediately evident from the provided output snippets alone.
- **Overall Comparison:** Based on the provided evaluation metrics, Random Walk, DIAMOnD, and GCN demonstrated the strongest overall performance in terms of retrieving a larger proportion of known disease proteins within their ranked lists (highest Recall@k and AP). However, Random Walk and DIAMOnD's score distributions (especially DIAMOnD's uniform top scores) make

prioritizing within the very top predictions challenging. The Neighborhood algorithm, while having lower overall performance metrics in this instance, provided a more interpretable score distribution with strong differentiation among top candidates and successfully identified a known key gene with a perfect score, highlighting its strength in identifying highly confident, directly relevant candidates. The Node2Vec algorithm performed poorly across all metrics.

Based on these results, Random Walk, DIAMOnD, and GCN appear most effective for maximizing the number of true positives identified at various ranks, while the Neighborhood algorithm is particularly strong at identifying highly connected, potentially well-known disease-associated proteins with high confidence scores. The choice of the "best" algorithm may depend on whether the goal is broad discovery (higher recall) or identification of highly confident, directly relevant candidates.

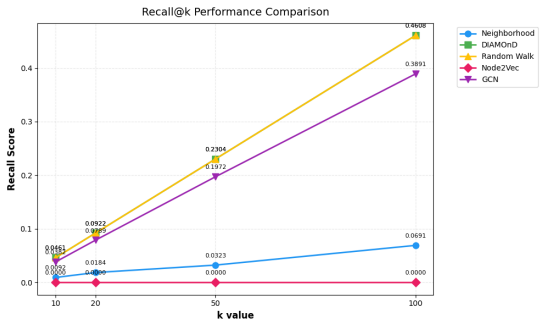

Fig. 9. Performance Comparision of Recall@k Metrics.

A. Community Structure Properties

We also analyzed the community structure properties of the disease pathway within the network using metrics described in Section IV-B.

```
disease_proteins: 217
disease_clustering: 0.1478
avg_disease_degree: 98.1613
largest_component_size: 177
```

disease proteins: 217

Total number of proteins (genes) associated with Liver carcinoma These are known proteins involved in the disease disease clustering: 0.1478

Clustering coefficient of disease-associated proteins Measures how well disease proteins are connected to each other Value of 0.1478 indicates moderate clustering (range 0-1) Shows disease proteins form some local communities but aren't densely connected avg disease degree: 98.1613

Average number of connections per disease protein Each disease protein connects to 98 other proteins on average High value suggests disease proteins are hub nodes in the network largest component size: 177

Size of the largest connected group of disease proteins 177 out of 217 disease proteins form one large connected component Indicates most disease proteins (82 percent) are part of the same network module These metrics suggest that Liver carcinoma proteins form a moderately connected module with significant interactions but not complete connectivity.

VII. Code Availability

The source code for the algorithms and analysis presented in this paper is publicly available on GitHub at the following repository:

https://github.com/atishay08/SIN-Project

VIII. Conclusion

This project explored the application and evaluation of several network algorithms for predicting disease-associated proteins within a Human Protein Interaction Network. We implemented and compared the performance of Neighborhood, DIAMOnD , Random Walk, Node2Vec, and GCN algorithms using standard evaluation metrics.

Our analysis revealed variations in the performance and characteristics of these algorithms. Notably, the Neighborhood algorithm, despite its simplicity, demonstrated robust performance in our specific prediction task, exhibiting strong score differentiation and effectively identifying known disease-relevant proteins. This highlights that for certain network structures and prediction tasks, straightforward connectivity-based methods can be highly effective. While other algorithms like DIAMOnD, Random Walk, Node2Vec, and GCN offer different theoretical advantages and approaches to network analysis, their performance in this specific study, as measured by our evaluation metrics and score analysis, did not outperform the Neighborhood method.

The observed differences in algorithm performance underscore the importance of evaluating various methods for a given task and dataset, as the optimal approach can be dependent on the specific network characteristics and the nature of the biological problem being addressed.

The study of Protein-Protein Interaction networks shares significant conceptual and methodological parallels with the analysis of social and information networks. All are types of complex networks where the relationships between entities (proteins, people, documents) dictate system-level behavior. The algorithms and techniques employed in this report, such as Random Walks, Node2Vec embeddings, and Graph Convolutional Networks, have direct counterparts and origins in the methods used to understand and predict interactions, spread of information, and community structure in social media, communication networks, and the web. Challenges faced in analyzing fragmented disease pathways in biological networks are analogous to identifying dispersed communities or influential but not necessarily centrally located nodes in social graphs. Thus, the computational framework and the insights gained from evaluating network algorithms in this biological context contribute not only to bioinformatics but also enrich the methodologies and understanding applicable to social and information networks.

A. Future Work

- Applying these algorithms to a wider range of diseases to assess the consistency of performance.
- Exploring hybrid approaches that combine the strengths of different algorithms.
- Incorporating additional biological data (e.g., gene expression, functional annotations) into the prediction frameworks.
- Optimizing the hyperparameters of each algorithm to ensure peak performance for the specific dataset and task.

REFERENCES

[1] "Large-scale analysis of disease pathways in the human interactome."
[2] "International Statistical Classification of Diseases and Related Health Problems Tenth Revision."
[3] "A DIseAse MOdule Detection (DIAMOnD) Algorithm Derived from a Systematic Analysis of Connectivity Patterns of Disease Proteins in the Human Interactome."
[4] "Modules, networks and systems medicine for understanding disease and aiding diagnosis."
[5] "Link prediction and classification in social networks and its application in healthcare and systems biology."
[6] "Social Networks Benchmark Dataset for Diseases Classification."
[7] "Social Networks, Risk-Potential Networks, Health, and Disease."
[8] "Disease Ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data."
[9] "https://snap.stanford.edu/biodata/datasets/10005/10005-D-DoPathways.html"
[10] "https://snap.stanford.edu/pathways/"