Faculty of Engineering
Cairo University

Bioinformatics Project

Team 3

Idea #3 "Biological Networks"

Submitted to Dr. Ibrahim Youssef

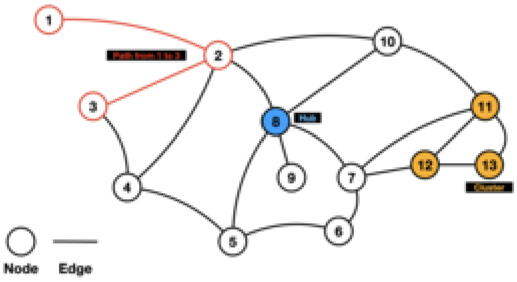
Name	ID
Aya Amr	9202342
Mahmoud Rabea	9203396
Mostafa Mahmoud	9212962
Hanya Ahmed	9203663

-

Table of Contents

Introduction

Biological networks are considered an efficient way to represent different interactions and relations between various biological entities mostly represented using graphs.

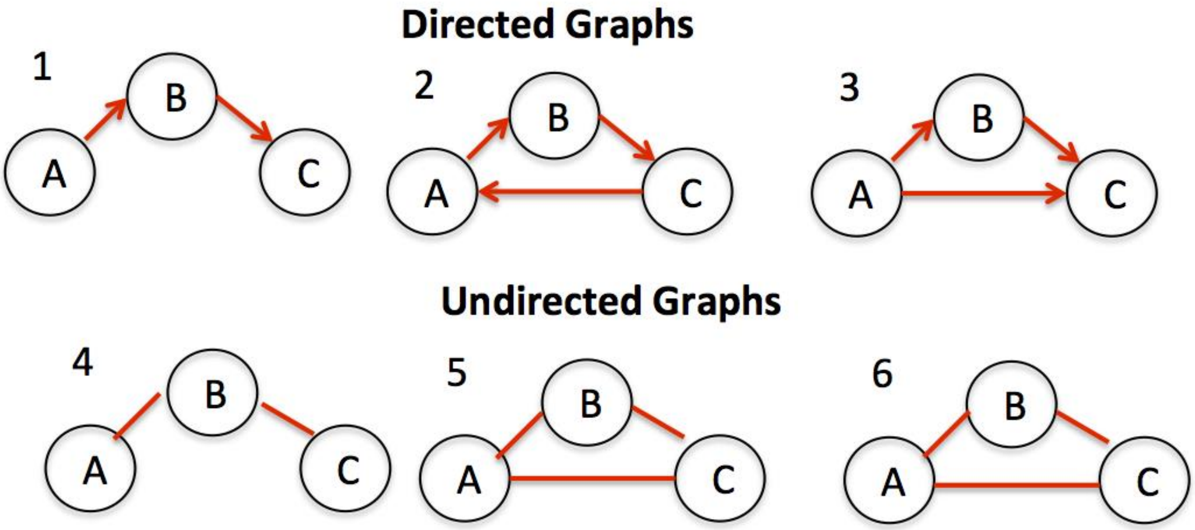


https://en.wikipedia.org/wiki/Biological_network

Graphs are mathematical objects consists of Vertices - donating the elements - and edges - donating the connections of these elements -

graph types:

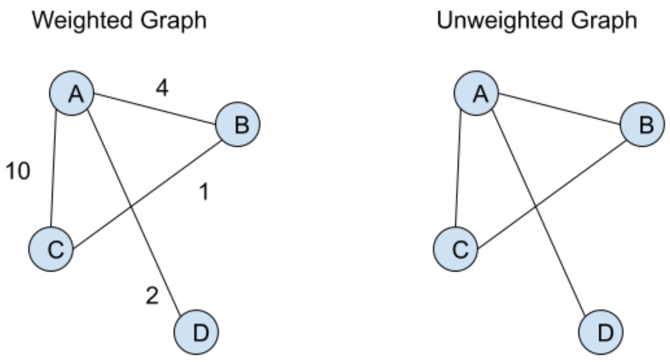
- Directed graph : each vertex has in degrees and out degrees connections which specify the relation between this vertex and other vertices in the network
- Undirected graph : in&out degrees are not taken in consideration



directed-undirected graph

- Weighted graph: each edge has a weight from 0 to 1 donating the interaction level between the two connected vertices.

- Unweighted graph: edge weights are not taken in consideration



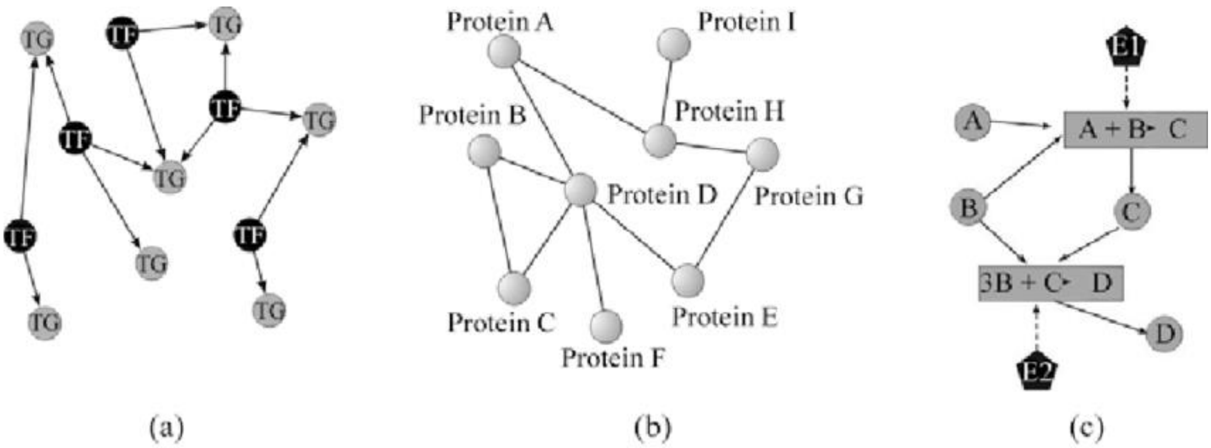
Weighted-Unweighted graph

Biological networks can be divided into three main categories:

a- Transcriptional Regulatory Network: consists of transcriptional factors TF and target genes.

b- Protein-Protein Interaction Networks (PPIs): each interaction between proteins specifying certain function for the cell.

c- Metabolic network: considers chemical reactions and enzymes.



The topic of our Project is PPIs networks and how to study, manipulate and visualize them.

Methods

This project was mainly implemented using the following python packages:

1- NetworkX : the main package to study, organize and manipulate biological networks

2- Pandas : For reading and handling given data

- Note that the dataset used is “PathLinker_2018_human-ppi-weighted-cap0_75.txt” found here

3- Matplotlib : for visualization and plotting

4- Numpy : for computations

5- Unipressed: Python client for the Uniprot API

Our project can be divided into 7 main points as follows:

1- Visualize and save the weighted graph for the given dataset and save the first 100 proteins of highest degree centrality

2- Given two proteins: list and save using NetworkX and matplotlib
Dijkstra’s algorithm shortest path (weighted and unweighted)
All shortest paths

3- Given one protein: list and save the protein’s connections specifying it’s in and out degrees using NetworkX and pandas

4- Given a set of proteins: list and save the degrees of the set descendingly and plot the frequency of the degrees using matplotlib and pandas

5- Convert the Protein’s UniProtID to its corresponding gene name using unipressed and save the results in a text file

6- Convert the Weighted graph of the given data into Unweighted graph and saving both the unweighted graph and its adjacency matrix as .txt using NetworkX and Matplotlib

7- Creates and save the minimum spanning tree of the given data using NetworkX and Matplotlib.

Results and Discussion

Upon constructing a biological network of the given interactome, it was observed that a graph of all listed interactions (over 0.5 million) required heavy processing and was visually unfathomable. Therefore, it seemed appropriate to find and store the highest hundred proteins in degree centrality, as they indicate the most influential proteins in the given list. These hundred proteins were saved in a descending order in a csv file (Top100DegreeCentralityProteins.csv)

Furthermore, by studying the highest five of these hundred proteins with respect to their in-degree centrality, the first protein was found to be P05067. This protein maps to a gene that performs physiological functions on the surface of neurons relevant to neurite growth, neuronal adhesion, and axonogenesis. The second protein, P04629, maps to a gene that is involved in neurotrophic-binding and specifying sensory neuron subtypes. These results seemed comprehensible, as such proteins that play a vital role in the nervous system are expected to interact with many proteins to achieve required physiological functions. The third protein, Q15717, is an RNA-binding protein, while the fourth protein, P01106, is a DNA-binding protein. The fifth protein, P05412, is a transcription factor.

Moreover, we constructed a biological network of a sample of around 2500 PPIs, and this centrality property was further confirmed, as the resulting graph showed what we might call central proteins with a high degree and peripheral proteins with a low degree: mostly of one.

When finding the shortest acyclic paths between two proteins: Q8TBF4, and P55157, two methods were tried. The first was finding the unweighted shortest path, then calculating its weight, and the second was finding the weighted shortest path. We used Dijkstra’s algorithm in both cases. As expected, we found that despite the first method yielding a path with fewer number of nodes; four, its calculated weight proved to be higher than that of the second method’s output, although it has a larger number of nodes; seven, in that case. In order to visualize all shortest paths, a minimum spanning tree was also constructed.

At the next stage of our analysis, we picked the most common protein, P05067, and listed all its directly connected proteins and the weight of each interaction in a text file (protein_connections.txt), in addition to its in and out degrees, which were almost equal with a small margin, listed in the last two lines of the text file.

Moreover, by taking a set of proteins from the given protein list, it was important to study the most frequent degrees among these proteins. We considered the in-degree, and by plotting its histogram, it was found that the most frequent degrees were the smallest. Considering the previous findings of protein degree centrality, this was coherent with our analysis. We created a text file of the set of proteins with their in-degree listed in a descending order (sorted_setOfProteins.txt)

The final step of our analysis was constructing the unweighted graph of the interactome and saving its adjacency matrix to a text file (adjacency1.txt)

Conclusion

Our PPI analysis visualized the centrality of PPIs around some more influential proteins, which was further proved by the histogram and ranking of proteins according to their in-degree. It has also shown that these more influential proteins play vital roles in achieving physiological functions, which require heavy interactions with other proteins. It has also proved that the shortest path of a weighted graph is not dependent on the number of nodes, but on the accumulative weight of that path.

Members Contribution

Name	Contribution
Aya Amr	Points #2 and #6, Presentation, Web App
Mahmoud Rabea	Points #4 and #5, Report, Web App
Mostafa Mahmoud	Points #1 and #6, Presentation, Web App
Hanya Ahmed	Points #1 and #3, Report, Web App

-

References

Multimeric, “Multimeric/Unipressed: Comprehensive Python client for the Uniprot Rest Api,” *GitHub.* https://github.com/multimeric/Unipressed

Minimum Spanning Tree, *Wikipedia*, https://en.wikipedia.org/wiki/Minimum_spanning_tree

“Protein protein interaction,” *Protein Protein Interaction - an overview* | *ScienceDirect Topics*. https://www.sciencedirect.com/topics/immunology-and-microbiology/protein-protein-interaction

https://www.ncbi.nlm.nih.gov/gene/4914

https://www.genecards.org/cgi-bin/carddisp.pl?gene=APP

https://www.uniprot.org/uniprotkb/P01106/entry

https://www.uniprot.org/uniprotkb/P05412/entry

https://www.ebi.ac.uk/training/online/courses/network-analysis-of-protein-interaction-data-an-introduction/introduction-to-graph-theory/graph-theory-adjacency-matrices/

-