Analysis of interactions in the S.cerevisiae Proteome

Nikolaos Gialitsis
Data Analytics

April 2019

Background:

Proteins are biological macromolecules that play a crucial role to all functions in living organisms. Studying how they interact , might provide insight that is beneficial to curing diseases and designing drugs. One of the most important model organisms used in research concerning the development and the function of the eukaryotic cell , is Saccharomyces cerevisiae , most commonly known as yeast. The yeast ‘s proteome , that is , the set of all proteins contained in yeast , forms a complex network of thousand interactions between it ’s components. Link analysis is a computer science field that tries to extract meaningful information from both artificial and natural networks. A relevant paper by Jeong, Hawoong & Mason [1] published by Nature magazine in June 2001 explored the correlation between the centrality and the lethality of the proteins , showing that proteins with higher degrees are more important for the survival of the cell than those with lower degrees. In this article, I aim to explore different ways to represent a protein’s importance and see how my results correlate with the previous findings , by applying methods from link analysis.

Introduction

A proteome can be modeled as a graph , with each node representing a different protein and an edge representing an interaction between two proteins. Since the flow of information is very hard to determine we assume that the network is undirected and that it can contain loops.
Centrality is a metric that can be used to assess a node’s importance in a graph. There are many ways to represent centrality ; in the next session I explore the following methods :

1. degree centrality
2. closeness centrality
3. betweenness centrality
4. pagerank centrality

I will also verify that the degrees in the protein network follow a power law with exponential cutoff as claimed by Jeong, Hawoong & Mason [1] by plotting the node degrees and fitting the values of the exponential function appropriately. For each method , the corresponding visual representations will be shown.

Last but not least , I will implement a propagation function which , starting from a given point S , scans the network and marks the nodes that can be affected by a mutation or deletion in S, either directly or indirectly. This can be done by a recursive search in the nodes that can be reached from S.

Methods

Following the object oriented principle, I created a class representing the graph network which contains all the necessary functions for link analysis.

Preprocessing:

First I read the information from a csv file , and I use a loop to estimate the number of lines that are included in the header , that is , the lines starting with the ‘%’ symbol that contain the file’s metadata (in this case header =2) .

Initialization:

I initialize the network by creating an instance of the Protein Network class:

```
G = ProteinNetwork(dataset)
```

which contains the following fields:

network : undirected graph implemented in the networkx library.
proteins : array containing all proteins in dataset (each protein = single number)
length : number of interactions between proteins (integer)
interactions : array of protein interactions , each interaction is a tuple : (protein1,protein2)
degrees : array of tuples (protein num, number of edges adjacent to protein)
Propagation_matrix : array containing all nodes that have been visited by a run of the color propagation algorithm
colored_nodes: the same as propagation matrix but in a format that can be interpreted by `np.asarray()`

I append each dataset's line as an edge in the network.

Analysis :

"How big is my protein network?"

The graph density 'D' of a graph can be calculated with the following formula and

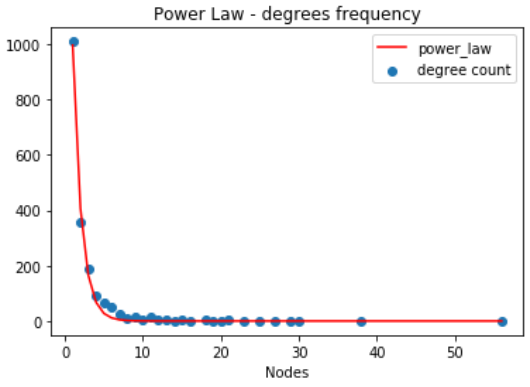
$$D = \frac{2|E|}{|V|\,(|V|-1)}$$

where |E| equals the number of edges and |V| the number of vertices (nodes) [2]

```
G.show_graph_info()
```

```
Graph density = 0.0013024208662
Number of edges : 2276
Number of nodes: 1870
```

The graph density of my graph is approximately 0.001 which means that the graph is quite sparse since it approaches the minimum density value which is 0.



"Which are the lowest and highest degrees in the network? How are degrees distributed along the Proteome? Which protein has the highest degree centrality"

The degree distribution in the network indeed seems to follow a power law as described in the paper [1]. That means that a lot of proteins have a few interactions and a very small percentage has many interactions. The red line represents the power law function that optimally fits the data and every blue

dot represents a protein. We can easily see that there exists a protein that has more than 50 interactions.

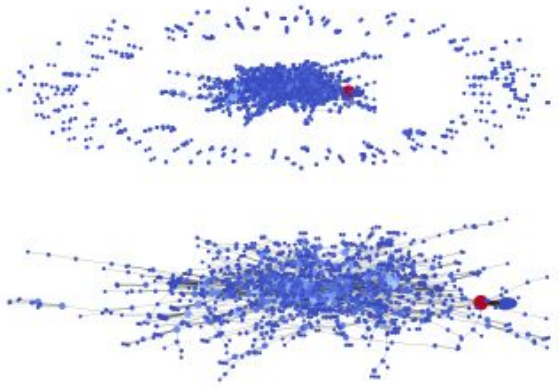
```
min degree =1
max degree =56
1010 nodes have minimum degree centralities
1 nodes nodes have maximum degree centralities
node with maximum degrees = 180 with num of degrees = 56
Propagation from point 180  affected 1458 nodes
```

In order to visualize our network we plot the graph with every protein shown as a dot. The dot's color goes from blue to red , depending on the number of interactions the protein has.

Also , the size gets adjusted so that proteins with higher degrees appear proportionally bigger.

The fist graph shows the whole network while the second graph shows the map of the proteins that can be affected by a change in the red (the dominant) protein , aka. the protein that has the 50+ degrees. It is easy to see that there exists a set of proteins that are independent , meaning that they are disconnected from the rest of the graph. In order to create the second graph I used the color_propagation algorithm with the starting point being the red protein.

The degree distribution of proteins



"Which proteins are the closest to most proteins in the Proteome? "

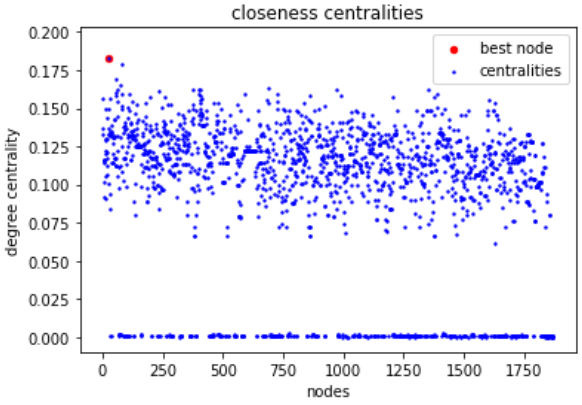
To answer this question , I implemented the show_closeness function where I calculate the closeness score for every protein. The formula for the closeness score is the following:

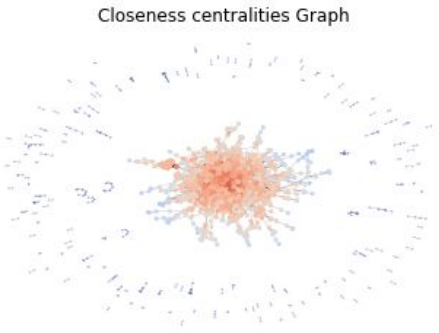
$$C(x) = \frac{1}{\sum_y d(y,x)} \quad [2]$$

where d(y,x) is the distance between proteins x and y. (aka. optimal number of edges needed to connect them).

```
24 nodes have minimum closeness centralities
1 nodes have maximum closeness centralities
```

The resulting graphs are the following:





Here we detect a different node being the most dominant than the one we found using the degree centrality measure. Also, now the difference between the values is larger since there exist proteins that have zero closeness centralities since there exists no paths to some nodes in the network. We also see that the proteins that are in the center of the network exhibit higher centralities than the ones in the radius.

“Which proteins are more likely to occur in a path between two proteins? “

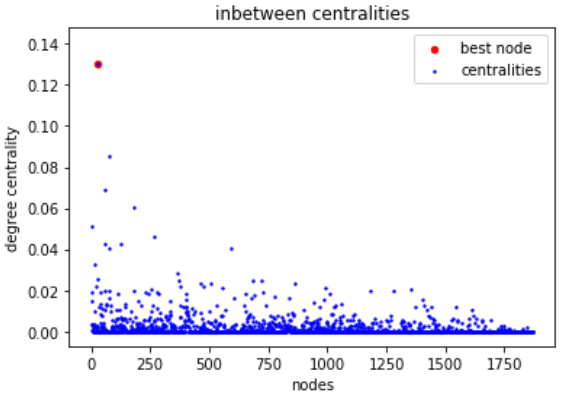
In order to answer this question, I calculated the in betweenness centralities using the show_betweenness function.

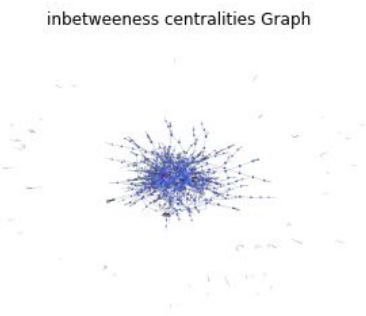
I discovered that the protein that has that centrality is the same as the one that has the highest closeness centrality. However the rest of the values seem to cluster close to 0 whereas for closeness centralities the values cluster around 0.125 . This is why the visualization of the network is not that big or colorful.The formula is the following: [2]

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v.

1124 nodes have minimum betweenness centralities
1 nodes have maximum betweenness centralities
max centrality node = 25 with value : 0.130445005365 and degree = 4

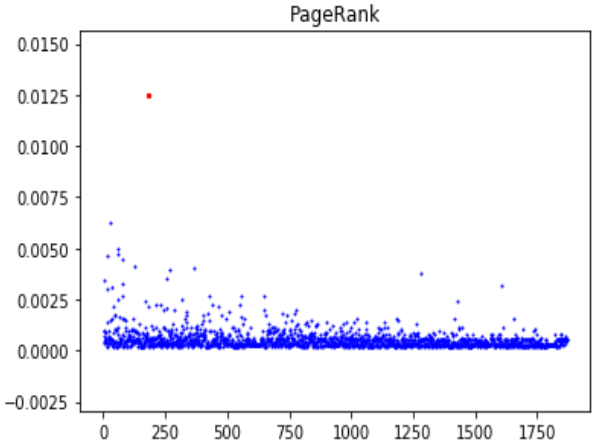


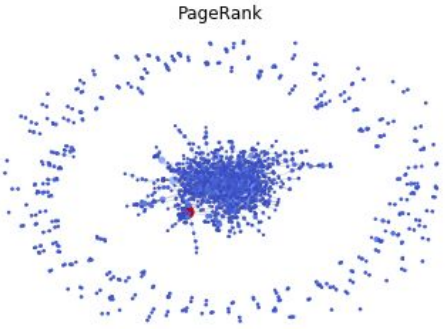


“**If a protein was webpage , which protein would show up first on a Google Search?”**

In order to answer this question I implemented the PageRank algorithm used to rank google's pages. The result is similar to the in betweenness centrality graph, but now most values are just above 0 , since the pagerank algorithm always gives some rank even to pages that get one or two votes (aka. proteins that have 1-2 interactions)

```
maximum page rank  = 0.0124715201311
best page rank node =  181
```





Conclusions

We see that there exist two layers in the network : the inner and the outer layer.
The inner layer is more dense and contains proteins that have only interactions with the other proteins in the same layer . On the other hand , the outer layer contains nodes that are only connected to a few other proteins in the outer layer as well and is much more sparse.

This means that the proteins in the outer layer do not affect as many other proteins but they are more robust to changes in the Proteome. On the other hand , a mutation or deletion in the inner layer has many implications. If a change happens in the most-connected proteins then the whole inner layer is affected.
I indeed observed that the degrees in the network follow a power law. However , the importance of the proteins from a link analysis standpoint , depends on the selected metric . For example , we have observed two dominant proteins , one of them has the highest betweenness , closeness and page rank centralities while the other one has the highest number of interactions.
Personally, I believe that judging the quality of a protein by just the number of degrees it has , is a naive approach since proteins are much more complex and not all interactions carry the same weight.

Citations:

[1] Jeong, Hawoong & Mason, S.P. & Barabasi, Albert-Laszlo & Oltvai, Z.N.. (2001). Lethality and Centrality in Protein Networks. Nature. 411. 41-2. 10.1038/35075138.
[2] https://en.wikipedia.org/wiki/Centrality