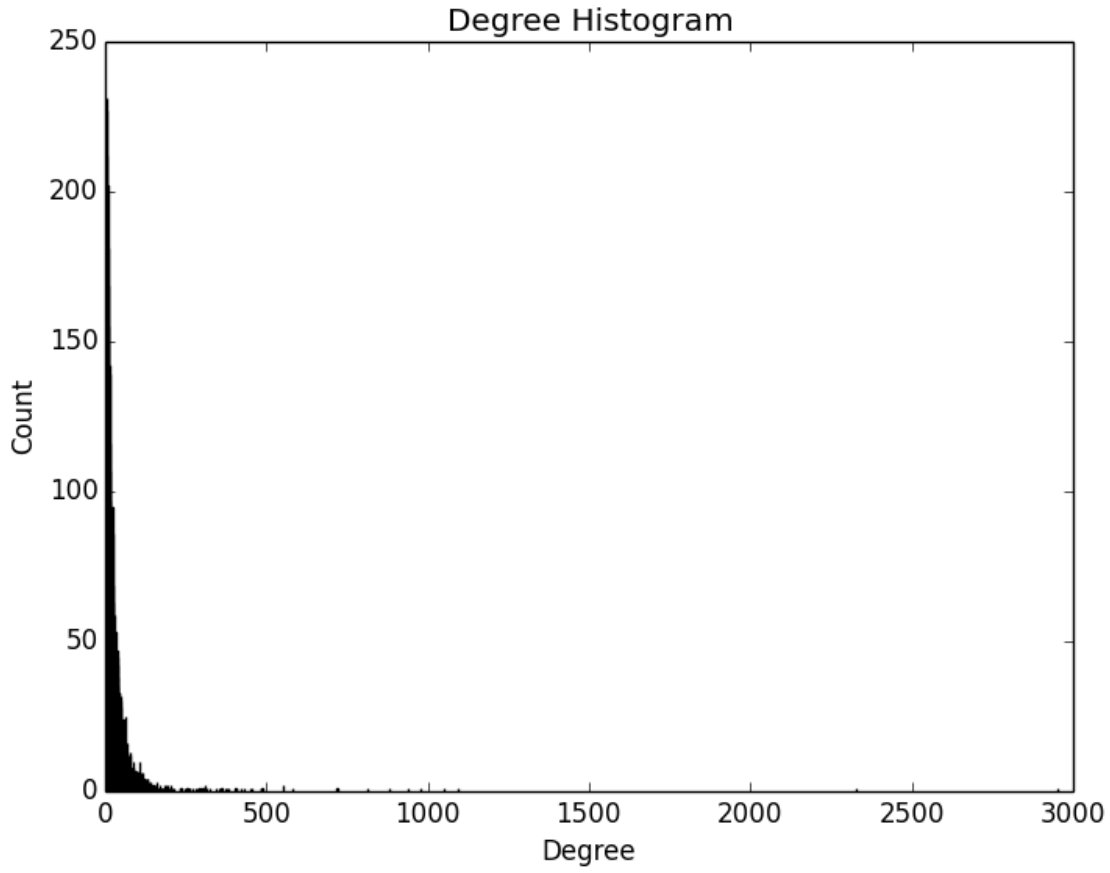
Written by: Aidan Fike
On: October 14, 2018
For: Comp167 Hw3

References: For most these problems, I used at least one function from the networkx package.
Additionally, to create a graph of the histogram, I copied some code from the following link.
https://networkx.github.io/documentation/stable/auto_examples/drawing/plot_degree_histogram.html

1a. I would argue that this network does indeed take the form of a scale-free network. As seen by the degree distribution, there is a clear pattern of there being many low-degree nodes and a handful of nodes with high connectivity. Additionally, the entire degree distribution takes the form of a 1/(x^gamma), as is characteristic with scale free networks (gamma is some small constant number typically between 0-10).

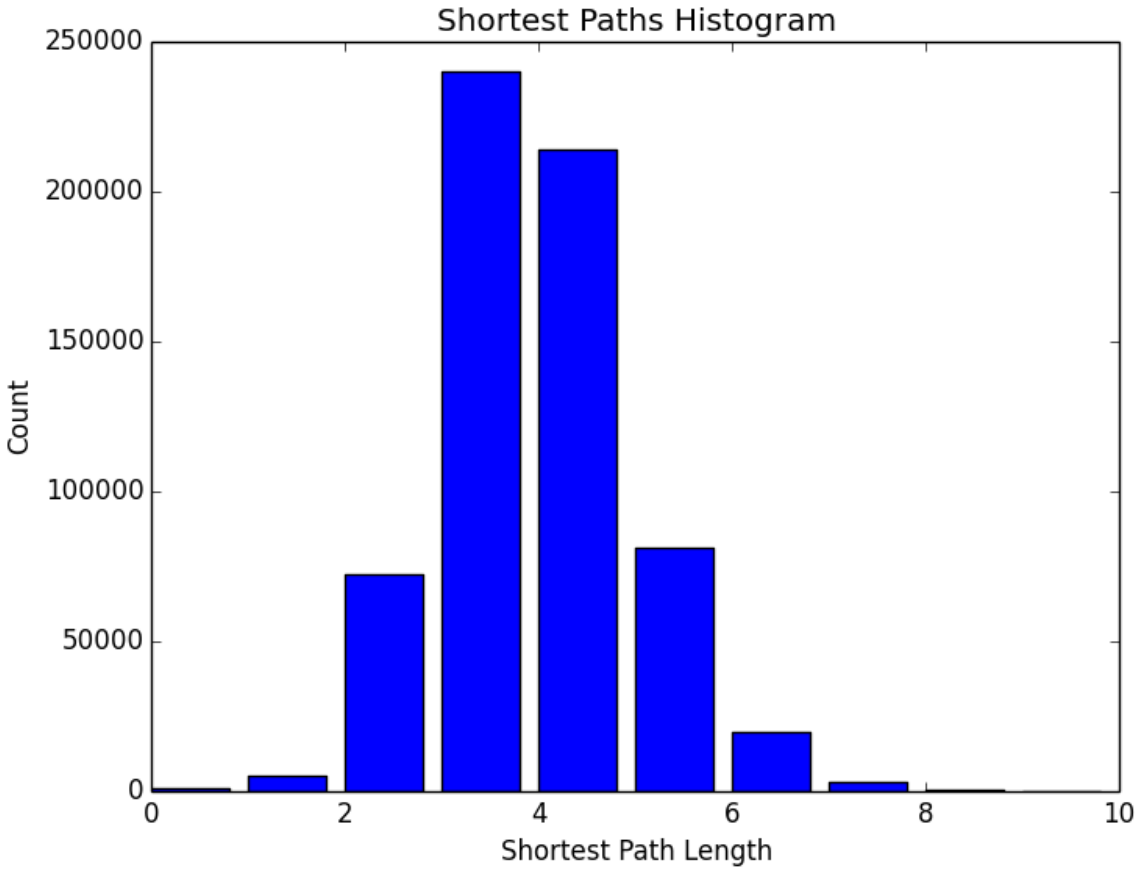


1b. Coefficient Coefficient: YGR296W and YPL098C both have 5 interacting

partners but YGR296W has a clustering coefficient of 0.1 while YPL098C has a clustering cofficient has 0.8. This difference between the two proteins must therefore lie in the number of links between the interacting partners that they contain. Specifically, because YPL098C has a cofficient that is 0.7 higher than YGR296W, we are able to conclude that the interacting partners of YPL098C interacted with each other more than the interacting partners of YGR296W do.

1c. I found 354514 3-cliques in this graph.

1d. The distribution oberseved does make sense for a protein-protein interaction network. Due to the scale-free nature of the graph, there are several proteins that interact with a large precentage of the potential proteins. Therefore, it makes sense that even though there are thousand proteins in this sample, the shortest path between them was, on average, roughly 3. Most of these paths likely went through one of the nodes with of a significantly large degree



1e. Viewing the shortest paths distribution, it can be seen that the longest path length found is roughly 7, indicating that if this sample of nodes is representative, the diameter of the graph is also 7. However, it is very

possible that adding more nodes could introduce even shorter possible paths, or, alternatively, introduce nodes that require even longer paths between them. Therefore, it is hard to say with certainty that this 7 diameter is representative for the entire ppi network.

2b. In my improved algorithm, I took into account the confidences of each edge that were given in the inital ppi file. For the non-improved algorithm, potential labels were collected from the neighbors, and labels which appeared the most was guessed to be the "true" label of that protein. In my improved algorithm, instead of only taking into account the number of times a given label appears, I now also take into account the confidence of the edge between the source node and the neighbors with a given label. Specifically, when a given label is found at a neighbor, instead of simply adding 1 to the weight of the label, the confidence of the edge between the desired node and its neighbor was added. This made is so that when a label was found at a neighbor through an edge with a lower confidence, that label was proportionatily penalized compared to a label that exists through an edge with a higher confidence.

The benefit of this is that edges with low confidence matter less than edges with higher confidence. By using information about confidence in the calculation, you are able to avoid relying on, for instance, data which is very noisey or data which may have been obtained using unreliable methods.

This change made a signficiant difference. For instance, when testing YDR143C I found that with the non-confidence method chose the label 16 with a weight 27 whereas with the condifence method, the label 1 was chosen with a weight of 16.25. This demonstrates how dangerous it is to weight each edge equally. If this is done, you are trusting faulty data just as much as reliable data.

2c. In fact, with this confidence-based scheme, the accuracy of this majority vote using leave one out cross validation improved from 0.466 to 0.507.

2d. Currently, to find potential labels for a given protein, I look at all its neighbors and weigh their labels based on the confidence of the edges with the test protein. If we desired to give multiple labels to a given node using majority vote, instead of only using the label with the highest weight, I would look at all the labels weights close to the highest-found weight. Then, I would label the desired protein with all of the labels which had weights within, for example, 10% of the highest weight found. In this scheme, if a given protein had 3 potential labels:
label: 45, weight: 9.7
label: 18, weight: 9.5
label: 23, weight: 4.6

The given protein would be given both the label 45 as well as 18 because 9.5 is between 90% and 110% of 9.7

To test the accuracy of this method, one possibility could be to simply extend the accuracy scheme used for a single label. Instead of testing a single label, we use leave-one-out cross validation to test the validity of all of the labels guessed for a given protein. For each label guessed for a given node, if that label appears in the trueLabel of that protein, we add a score of 1
Then, we divide the given node's score by the number of labels guessed to find the accuracy at the given node. Finally, to get the accuracy of the entire network, we average the accuracies of all of the nodes to get the overall accuracy of guessing labels.