Network of Diseases

Zidar Miha (63060317)

January 11, 2013

1 Introduction

In this homework assignment, we have build a network of deseases, using shared genes for determening connections between them. On top of that network we ran network clustering with label propagation. The key point was to see how and which deseases are connected.

2 Data

We were given an OMIM morbidmap file, that contained a list of deseases and their genes. The list of deseases contained multiple lines with the same desease name, so we had to filter the names and join the genes of simmilar names together. By doing that, we have reduced the number of deseases from 6238 to 2448. Because some desease names contain typos, we could reduce the latter number even more if we joined the deseases based on their levinshtein distance.

After we build the connected graph, we could see that the degree distribution follows the power law quite nicely (figure: 2)so we can assume that this graph is Scale free. The graphs are also quite well connected, since the diameter of the largest connected subgraph is only 12.

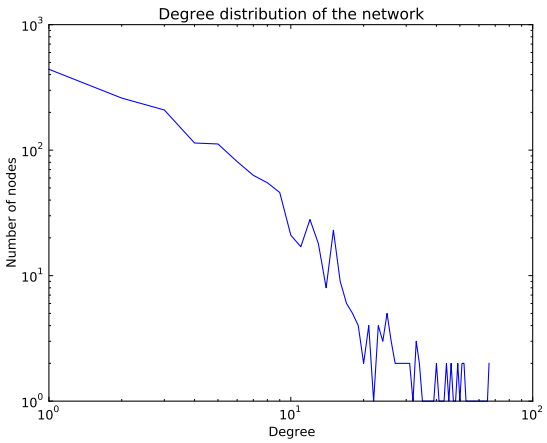

Figure 1: Degree Distributiona, in log-log scale

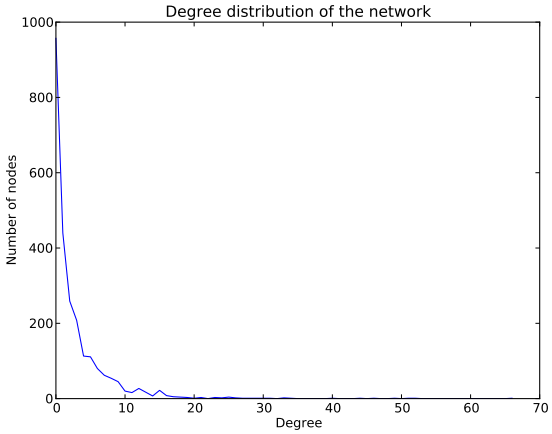

Figure 2: Degree Distributiona

3 Methods

For clustering we used label propagation algorithm by Raghavan. The algorithm starts with saying that each node is it's own cluster, and then, through a few iterations, it randomly walks over our graph, and updates cluster labels so that each node gets the most common label of its neighbours. If there are more than one such neighbours, then the algorithm should randomly select one of those. The method can run untill the labels are still changing, but I have limited it to a fixed number of iterations. The number of passes we did for the final result was 10000 label updates on the entire graph.

For displaying the graph, we used spring_layout, implement in pythons networkx librart. That algorithm tries to group the most connected graph points together and kep the distant points away from each other. beacuse this is always just approaching "the optimal" layout, it could run forever, so we set a hard limit of 2000 for the final result

```
clusters = {j:i for i,j in enumerate(g.nodes())}
for _ in xrange( iterations ):
    # list of all nodes
    nodes = g.nodes()
    random.shuffle(nodes)
    for node in nodes:
        # count all neigbour labels
        count = labelCounter(g.neighbors(node))
        # select radom candidate from the most common neigbours
        candidate = getCandidate(count)
        clusters[node] = candidate
```

4 Results

Our full graph has 1110 disjoint subgraphs, most of which are really small, and then we have just one big subgraph with 1118 nodes. The histogram (figure 3) shows the distribution of subgraph sizes, without the largest subgraph, because that would obscue the graph too much.

Below we have our final resutls of the clustering with the spring layout. Because there are so many nodes, the full view(figure 4) overlaps too much, we have a smaller slection (figure 5) which shows nice desease groupping.

Finally we have extracted clusters in which we have some more known deseases, to inspect what other deseases they relate to. The most surprising relation was between Diabetes, HIV and AIDS (figure 6).

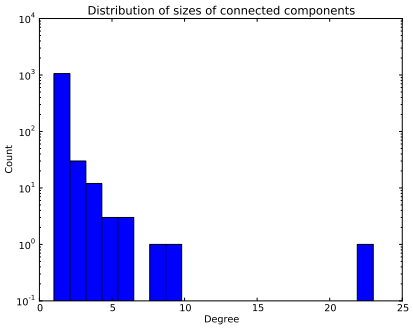

Figure 3: The distribution of sizes of connected subgraphs

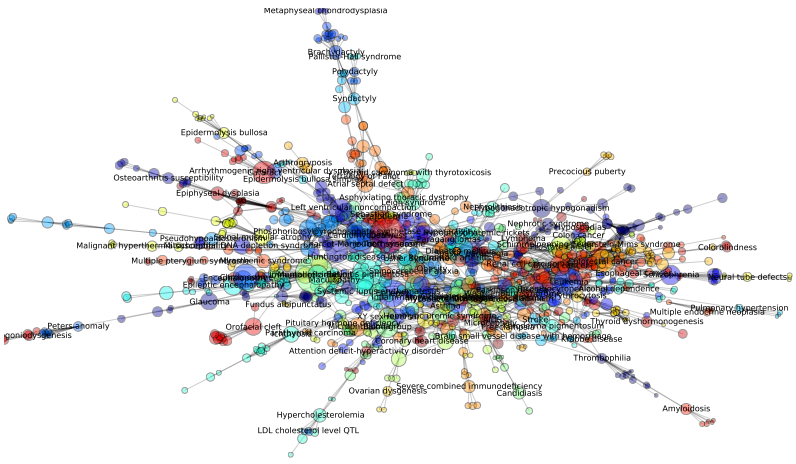

Figure 4: Entire graph of teh biggest connected component

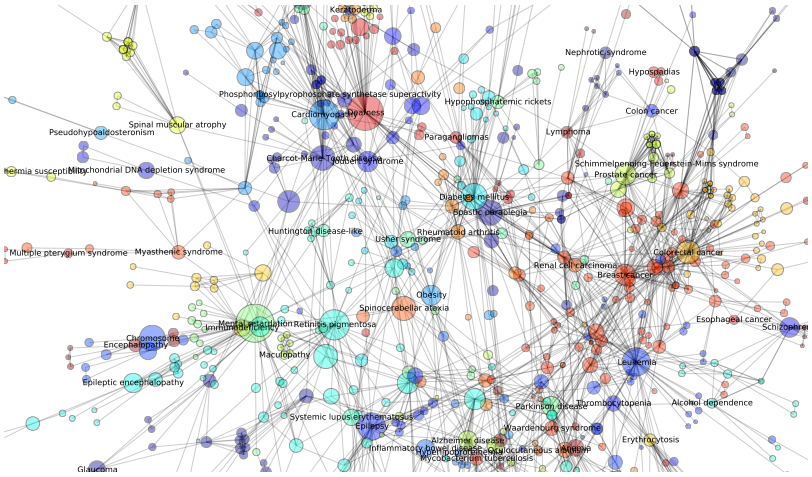

Figure 5: A smaller section of the graph of teh biggest connected component

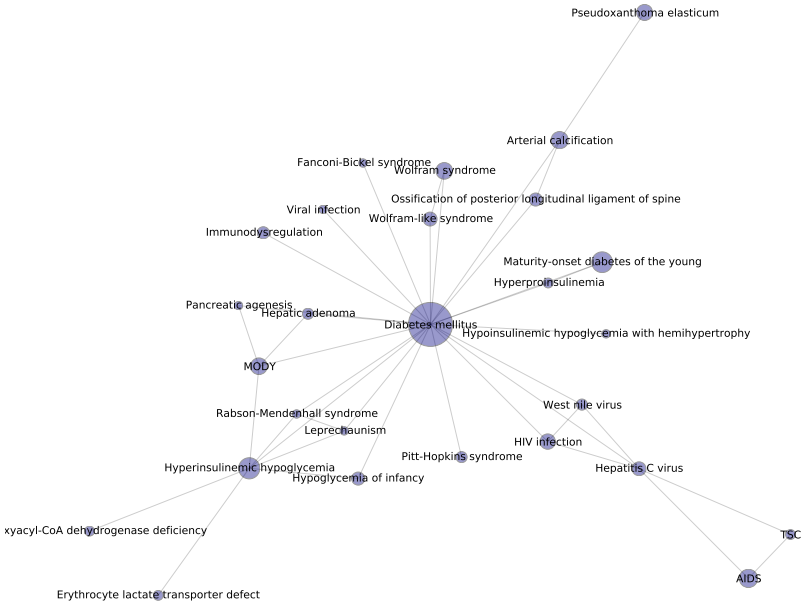

Figure 6: Diabetes mellitus cluster

Honor Code

My answers to homework are my own work. I did not make solutions or code available to anyone else. I did not engage in any other activities that will dishonestly improve my results or dishonestly improve/hurt the results of others.

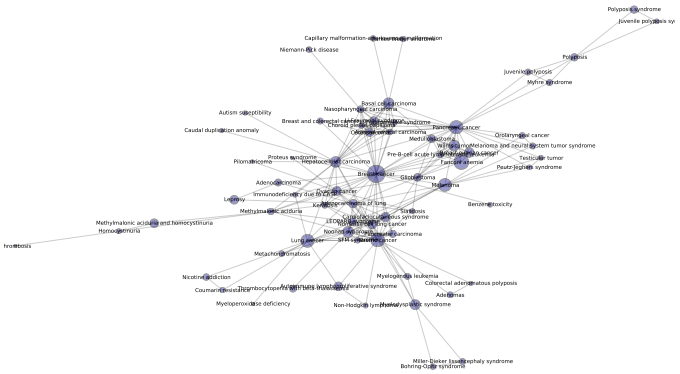

Figure 7: Breast cancer cluster

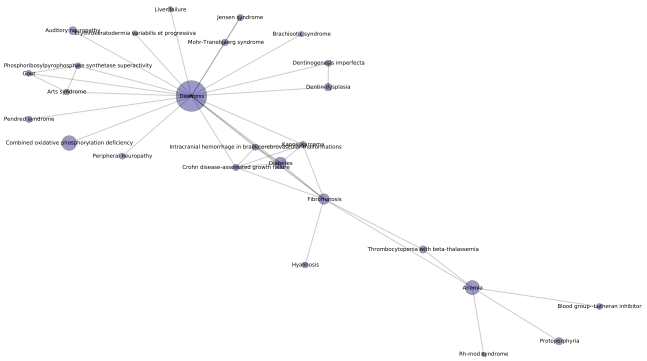

Figure 8: Deafness cluster