## Code explanation

To begin with, after reading data from the local cloud in Google Colaboratory, we had three functions defined to simplify the computation process. The first of three serves the purpose of computing centrality measures using predefined functions in NetworkX; the other two are for plotting and visualizing for better analysis; the first provides a heat map for each centrality measure, and the second one plots a histogram for each centrality measure.

Code Listing 1: Functions for future computations

```
def centrality_measures(g):
    d_cent = nx.degree_centrality(g) # this is normalized by default
    c_cent = nx.closeness_centrality(g) # this is normalized by default
    b_cent = nx.betweenness_centrality(g)
    e_cent = nx.eigenvector_centrality(g)
    pr = nx.pagerank(g) # alpha is 0.85 by default, we can change it.
    clus_coeff = nx.clustering(g)
    # these are all dictionaries from nodes to their centrality value (or other variables).
    return d_cent, c_cent, b_cent, e_cent, pr, clus_coeff

def draw(G, pos, measures, measure_name):
    nodes = nx.draw_networkx_nodes(G, pos, node_size=2, cmap=plt.cm.plasma ,
                                   node_color=list(measures.values()),
                                   nodelist=measures.keys())
    nodes.set_norm(mcolors.SymLogNorm(linthresh=0.01, linscale=1, base=10) )
    edges = nx.draw_networkx_edges(G, pos)

    plt.title(measure_name)
    plt.colorbar(nodes)
    plt.axis('off')
    plt.show()


def plot_hist(measure, measure_name):
    plt.hist(list(measure.values()))
    plt.title(measure_name)
    # plt.axis('off')
    plt.show()
```

## Dataset 1: General Relativity and Quantum Cosmology collaboration network

The first dataset was General Relativity and Quantum Cosmology collaboration network dataset or as it is referred to in the code: GrQc.

After trying to compute the radius, center, diameter, and eccentricity and getting errors giving infinity values, we discovered that our network was disconnected.

**Degree Centrality** Then, the defined functions were used to compute centralities. The first centrality is Degree Centrality. As shown in figure 1 histogram, most degree centralities are close to zero, and only a few are nearly 0.01. Since we are using a

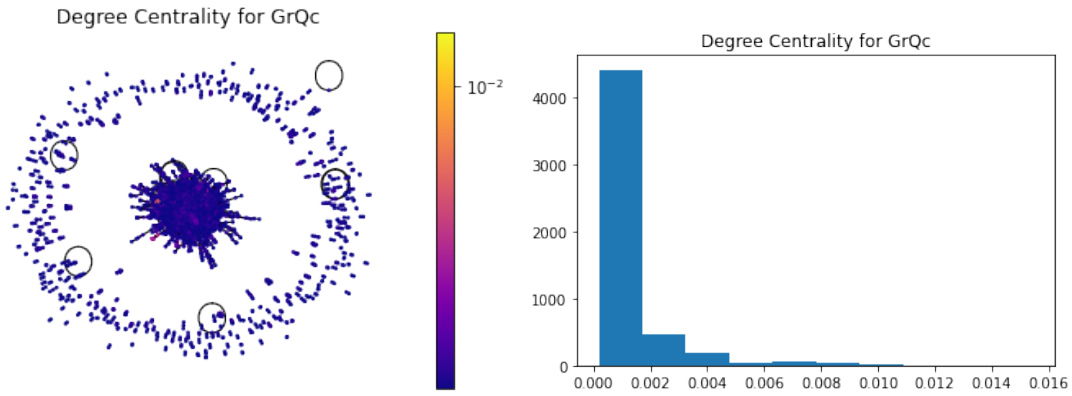


Figure 1: Heatmap for degree centrality and an approximate visualization of the network (on the left). Histogram of degree centrality of the nodes (on the right).

normalized measure, the highest possible centrality, meaning if we had a node connected to every other node, is 1. So we can conclude that our network is far from a complete graph, and we can say it's sparse.

**Closeness Centrality** The next measured centrality, closeness centrality, scores each node based on its sum of shortest paths. As seen in the figure 2, the distribution of closeness centrality is much more different from degree centrality. The histogram consists of some nodes whose closeness centrality is near zero and some with normal-like distribution with a mean near 0.14.
By looking at the heatmap, we can see what this means, some nodes in the middle are almost similarly close to each other, and there is a ring around these nodes with low closeness centrality meaning they're much further from each other than the ones in the middle.

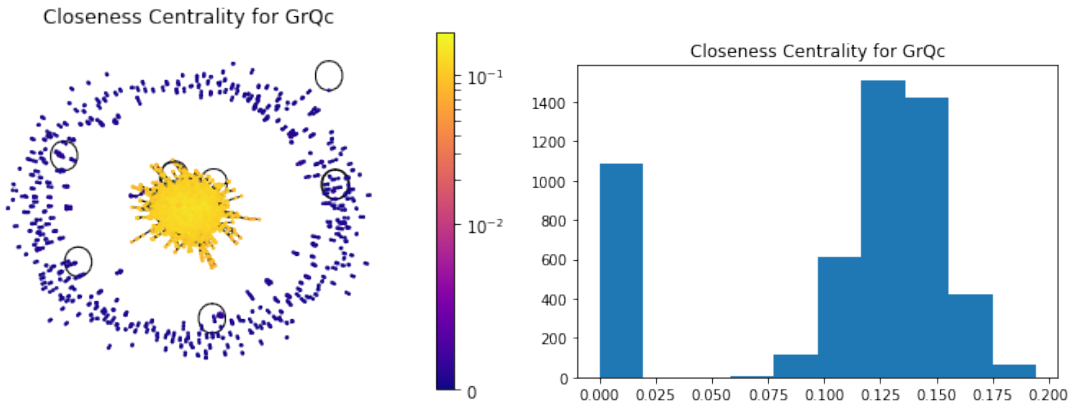


Figure 2: Heatmap for closeness centrality and an approximate network visualization (on the left). Histogram of closeness centrality of the nodes (on the right).

**Betweenness Centrality** counts the number of times a node lies on the shortest paths between other nodes. We can see in the figure 3 that the same thing happened to degree centrality is happening here as well. Most nodes have near zero betweenness centrality, and few are higher than 0.015. Combining the results from the last two measures, we

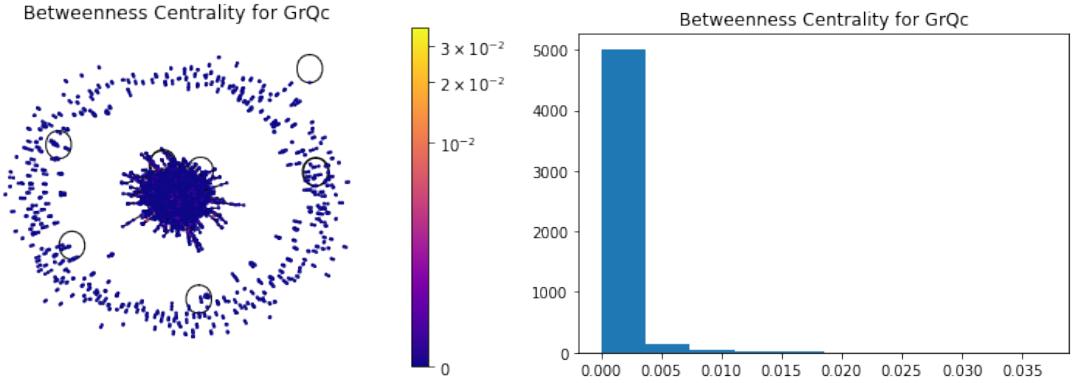


Figure 3: Heatmap for betweenness centrality and an approximate network visualization (on the left). Histogram of betweenness centrality of the nodes (on the right).

have some nodes that are relatively close to each other, but those same nodes are not on many shortest paths and are not important in this way.

**Eigenvector Centrality** can identify nodes with influence over the whole network, not just those directly connected to it. With this explanation, we could expect the output of the code, and seeing figure 4 confirms our expectations.

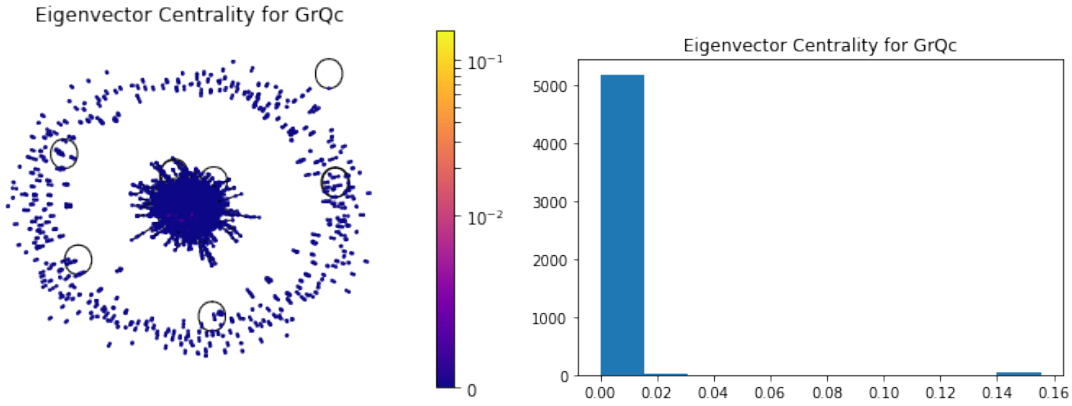


Figure 4: Heatmap for eigenvector centrality and an approximate network visualization (on the left). Histogram of eigenvector centrality of the nodes (on the right).

**PageRank** uncovers nodes whose influence extends beyond their direct connections into the wider network. This measure is useful in directed networks, and since our network is not directed, we may not be able to see this measure's full potential here. On

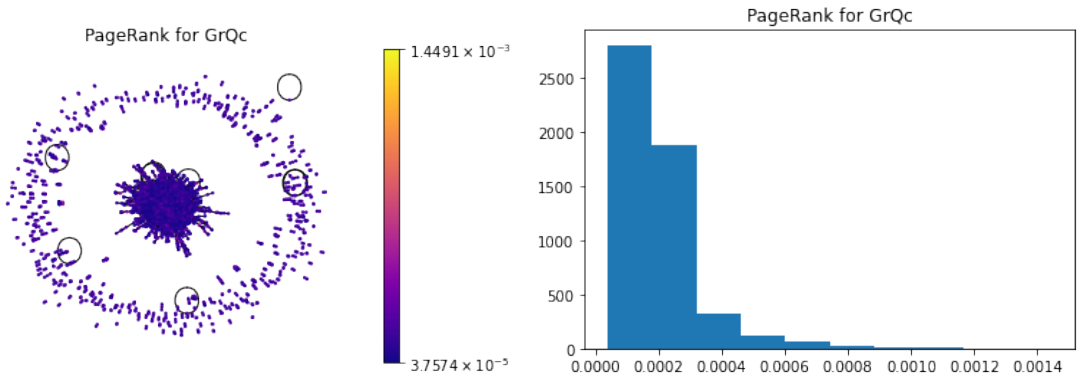


Figure 5: Heatmap for PageRank and an approximate network visualization (on the left). Histogram of PageRank of the nodes (on the right).

the histogram of PageRank, non-similar to betweenness and eigenvector centralities, its curve is not as steep as it was on former measures.

**Clustering Coefficient** And lastly, a clustering coefficient measures the degree to which nodes in a graph tend to cluster together. Looking at the heatmap in figure 6, we can see that nodes with clustering coefficient in the middle part are more than nodes with high cluster coefficient in the ring. Also, since there are not many connections on the ring, we can say that there are high clustering coefficients because some packs of fully connected nodes, such as two or three nodes connected, make their clustering coefficient

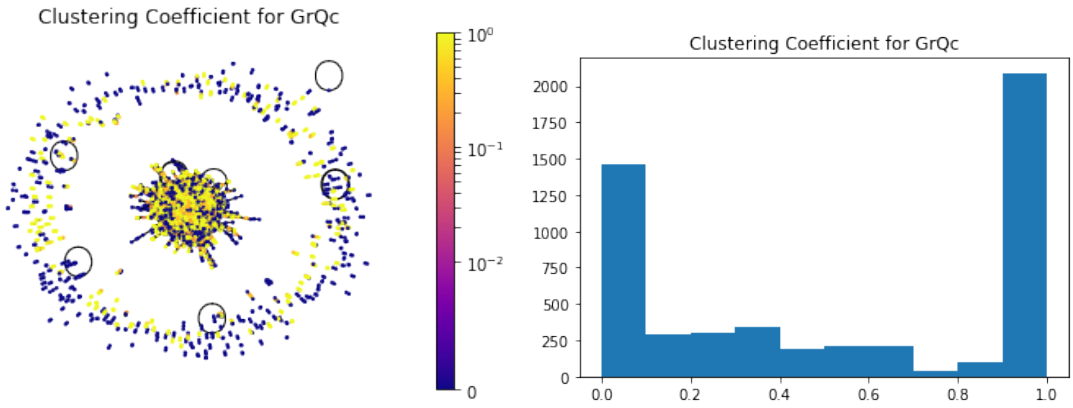


Figure 6: Heatmap for clustering coefficient and an approximate network visualization (on the left). Histogram of clustering coefficient of the nodes (on the right).

equal to one.

**Further measurements** There were some other measures computed, which are visible in the code, that are worth mentioning. We had 48,260 triangles, and with 5242 nodes, the total possible triangles were 23,993,367,880, meaning very few triangles were formed in the network. Moreover, there were bridges in the network (obviously), and the largest connected component was visualized in the middle of every figure up until now.
We had 4158 nodes in the largest connected component, which is close to %80 of the nodes, and we had 13428 edges in the largest connected component, which is around %92 of the total edges of the network. This means the nodes outside the largest connected component are barely connected, as we mentioned before.

### Dataset 2: Wikipedia vote network

This section's network is a directed network where a directed edge from node i to node j represents that user i voted on user j.