Computational Systems Biology (BT5240)

Assignment 2

Anirudh Rao (BE21B004)

Problem 1

To generate a **Watts-Strogatz network** with n nodes, k initial neighbours, and rewiring probability p, we first start with a **ring lattice** having n nodes, each of which is connected to $\frac{k}{2}$ neighbours on the left and $\frac{k}{2}$ neighbours on the right. The adjacency matrix for this can be obtained by performing **circular shifts** on the rows of an $n \times n$ identity matrix and summing these shifts. Circular shifting can be achieved in MATLAB using the inbuilt `circshift` function.

For example, a ring lattice with 6 nodes connected to 4 neighbours each can be obtained using:

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\texttt{circshift}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \texttt{circshift}_{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\texttt{circshift}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \texttt{circshift}_{-2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

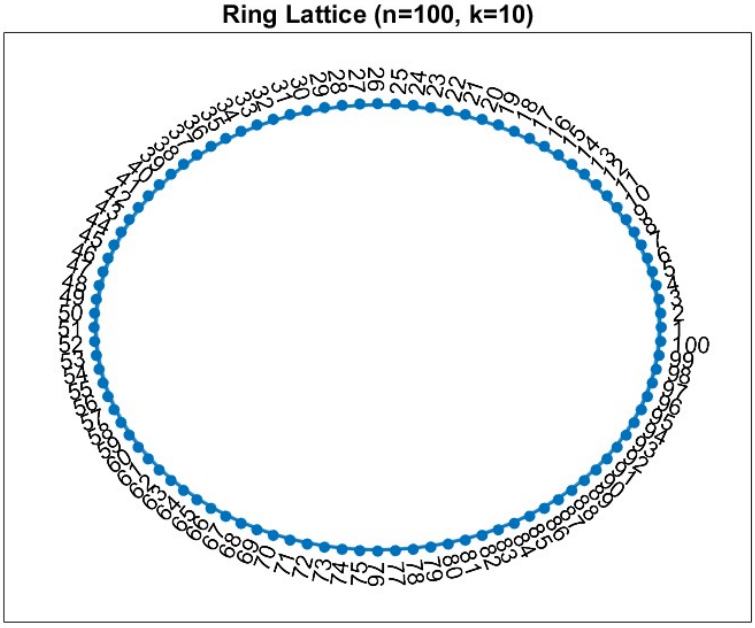
$$A_{\text{ring},n=6,k=4} = \texttt{circshift}_1 + \texttt{circshift}_{-1} + \texttt{circshift}_2 + \texttt{circshift}_{-2}$$

$$= \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

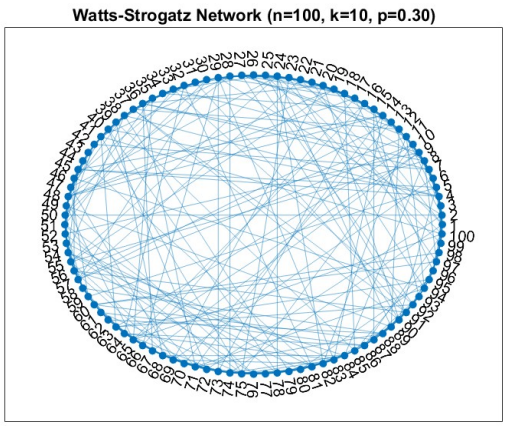
Once the ring lattice is obtained, we randomly **rewire** the existing edges with probability p to obtain the Watts-Strogatz network. To do this, we sample uniformly at random from $[0, 1]$ for every existing edge E_{ij} in the ring lattice. If the sampled number $< p$, we rewire E_{ij} such that it is still connected to node i but is now connected to a node j' that is sampled uniformly at random from the set of remaining nodes that are currently not connected to node i. Appropriate updates are made to the adjacency matrix of the network.

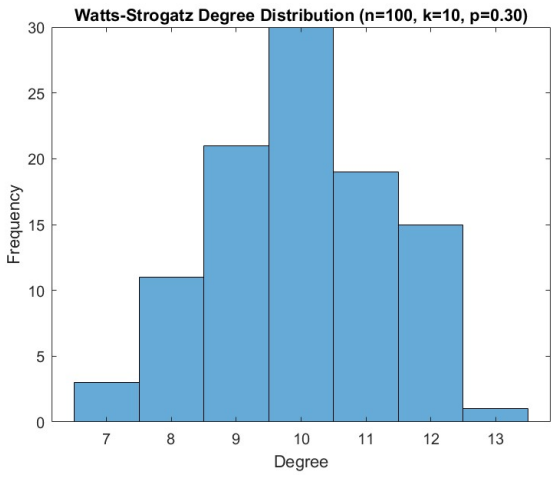
Based on the given problem, we set $n = 100$, $k = 10$, and $p \in [0.3, 0.7, 1]$.

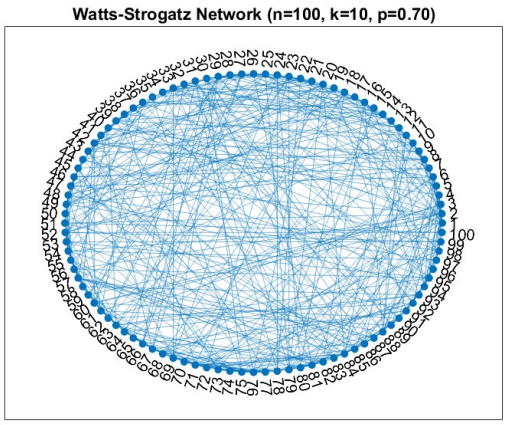
The original ring lattice obtained prior to rewiring is shown below:

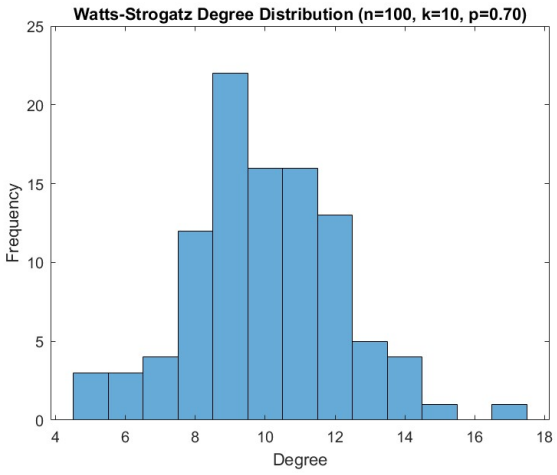


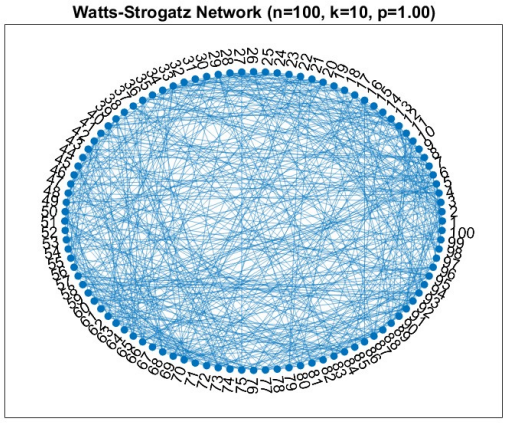
After rewiring, we obtain three different Watts-Strogatz networks. For each network, we calculate and plot the **degree distribution**.

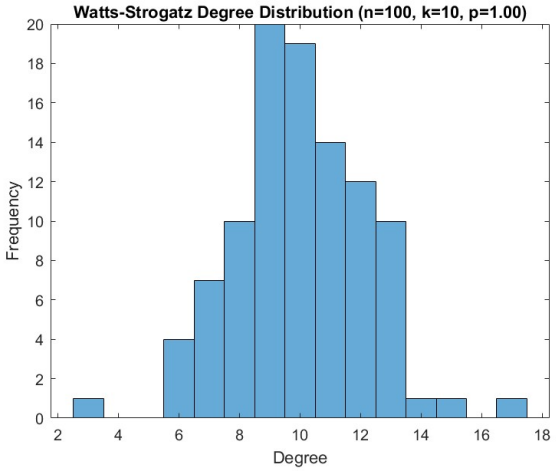












We then compute the **average clustering coefficient** and **characteristic path length** of each of the networks using functions available in MATLAB BGL.

p	Average clustering coefficient	Characteristic path length
0.3	0.262223	1.158400
0.7	0.109549	1.106500
1	0.082848	1.101500

To compare these networks against **Erdős–Rényi random networks**, we generate 100 different random networks with n nodes and probability of edge $\frac{n \cdot k}{2} / \binom{n}{2}$, and compute each of their average clustering coefficients and characteristic path lengths. We then find the mean (μ) and standard deviation (σ) of these parameters across the 100 random networks. These are found to be $\mu = 0.1009$ and $\sigma = 0.0096$ for average clustering coefficient, and $\mu = 1.1028$ and $\sigma = 0.0175$ for characteristic path length. Using these, we can perform a ***Z*-test** and compute the Z-score and p-value of the Watts-Strogatz network parameters to see if they are significantly different from those of the random networks. Using a Z-test is appropriate as the distribution of average clustering coefficient and characteristic path length across the 100 random networks is approximately normal.

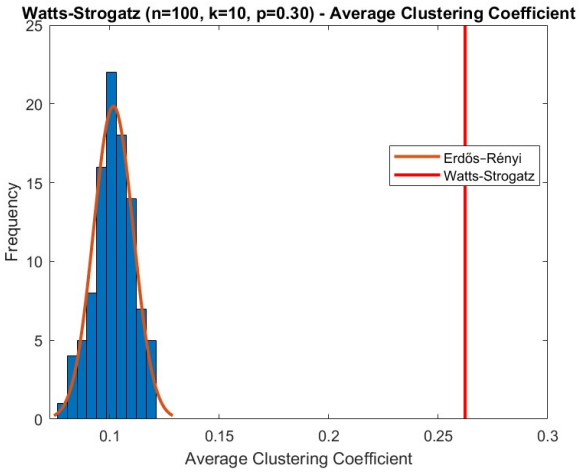
p	Average clustering coefficient	Z-score	p-value
0.3	0.262223	17.761976	0
0.7	0.109549	0.887651	0.374729
1	0.082848	−1.886162	0.059273

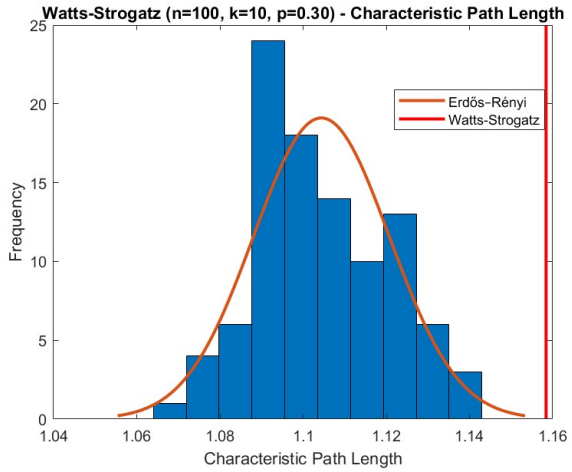
p	Characteristic path length	Z-score	p-value
0.3	1.158400	3.307800	0.000940
0.7	1.106500	0.251829	0.801173
1	1.101500	−0.076923	0.938685

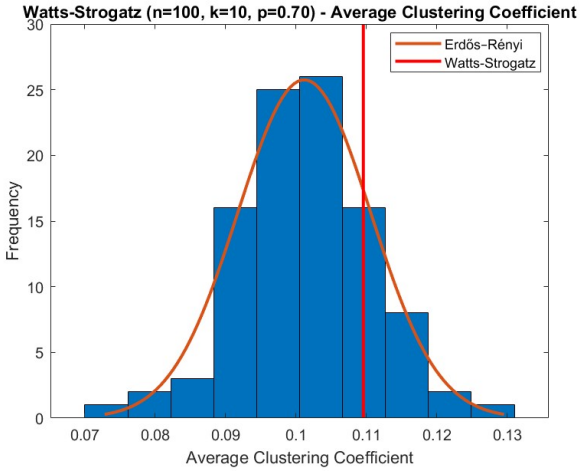
Using a significance threshold of 0.05, we can conclude the following:

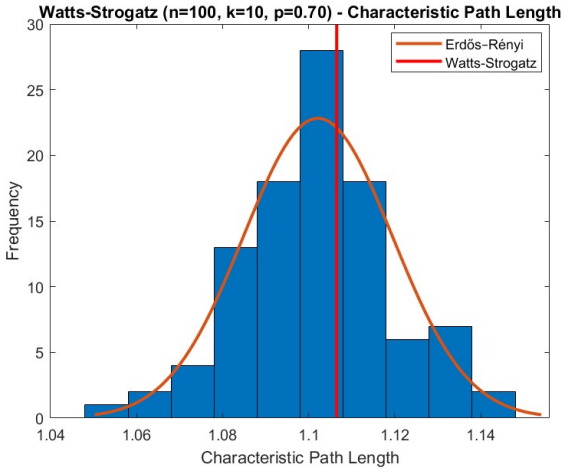
- When $p = 0.3$, the average clustering coefficient and characteristic path length of the Watts-Strogatz network are significantly different from that of random networks.
- When $p = 0.7$, the average clustering coefficient and characteristic path length of the Watts-Strogatz network are NOT significantly different from that of random networks.
- When $p = 1$, the average clustering coefficient and characteristic path length of the Watts-Strogatz network are NOT significantly different from that of random networks.

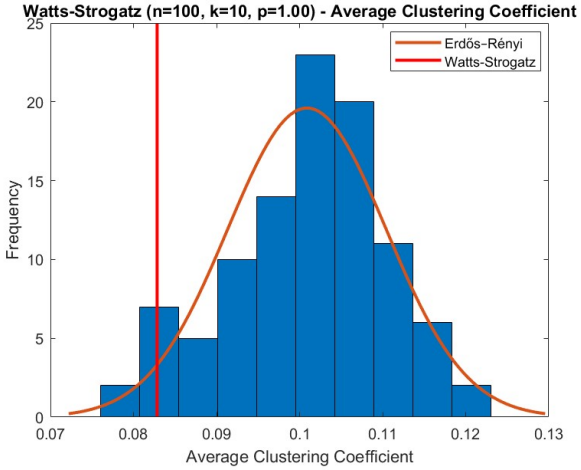
Lower the value of p, higher the average clustering coefficient and characteristic path length. As p increases, the Watts-Strogatz network becomes more and more random. At $p = 1$, the network becomes completely random.

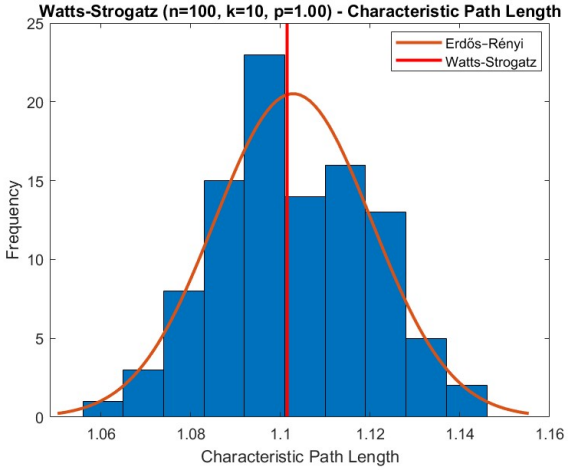










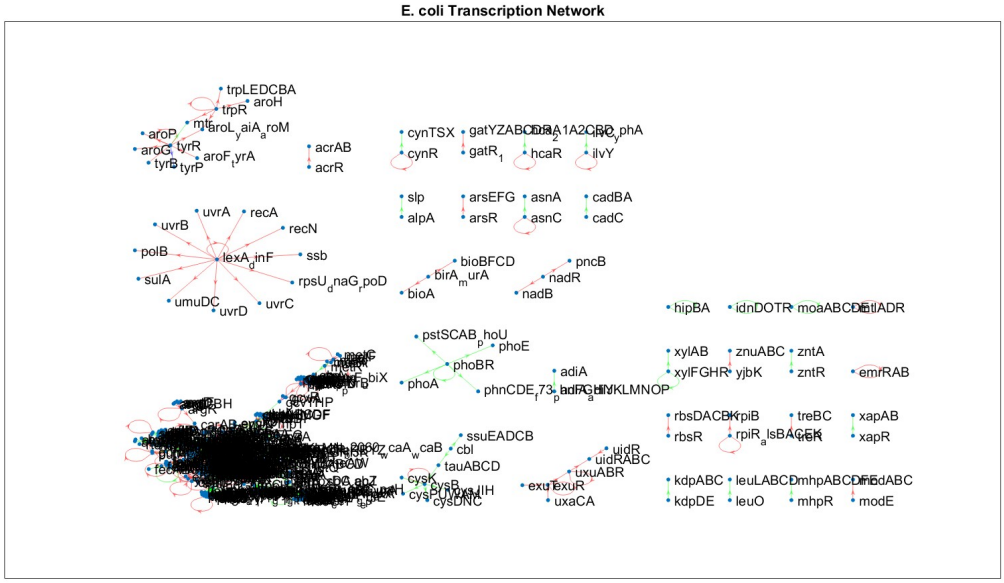


NOTE: Performing a Z-test in MATLAB requires the Statistics and Machine Learning Toolbox.

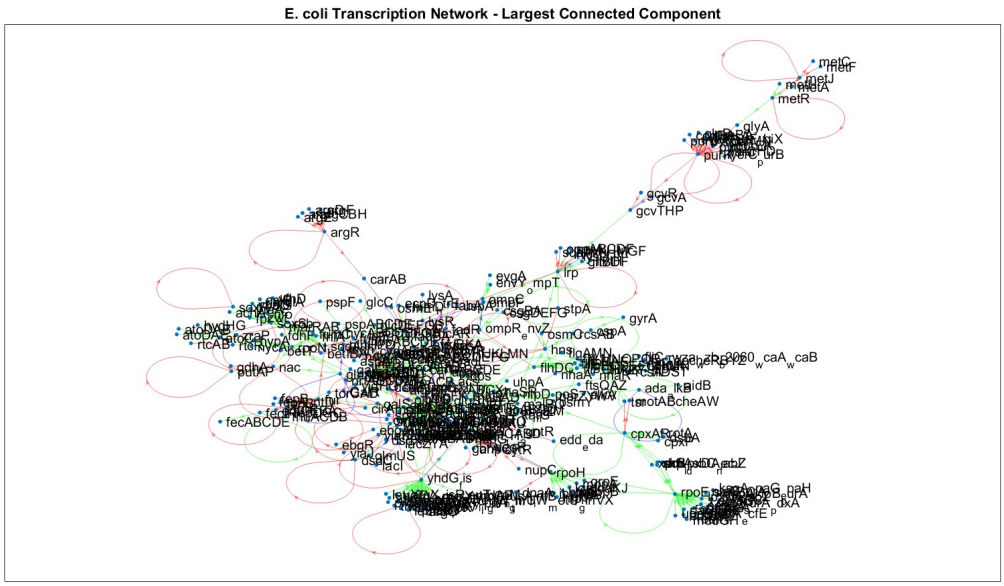
Problem 2A

The edge list of the *E. coli* transcription network is loaded into MATLAB as a table. This network is best represented as a **directed graph** as there is a notion of directionality in how a transcription factor regulates an operon.The network has 423 nodes and 578 edges.

We visualize this network and colour the edges according to the type of regulation - repression, activation, dual regulation.



The network is not fully connected and there are multiple connected components. The largest connected component is visualized below:



Problem 2B

We then compute the degree centrality, closeness centrality, and betweenness centrality of each node in the network, using the `centrality` function in MATLAB. The top 5 transcription factors according to each of these centrality measures is shown below:

Rank	Transcription factor	Degree centrality
1	crp	74
2	yhdG_fis	28
3	rpoE_rseABC	26
4	fnr	24
5	himA	23

Rank	Transcription factor	Closeness centrality
1	crp	0.00043481
2	rpoE_rseABC	0.00014753
3	fnr	0.00014743
4	yhdG_fis	0.000146
5	arcA	0.00011818

Rank	Transcription factor	Betweenness centrality
1	flhDC	49
2	fliAZY	47
3	rpoH	40
4	hns	22
5	ompR_envZ	18

Problem 2C

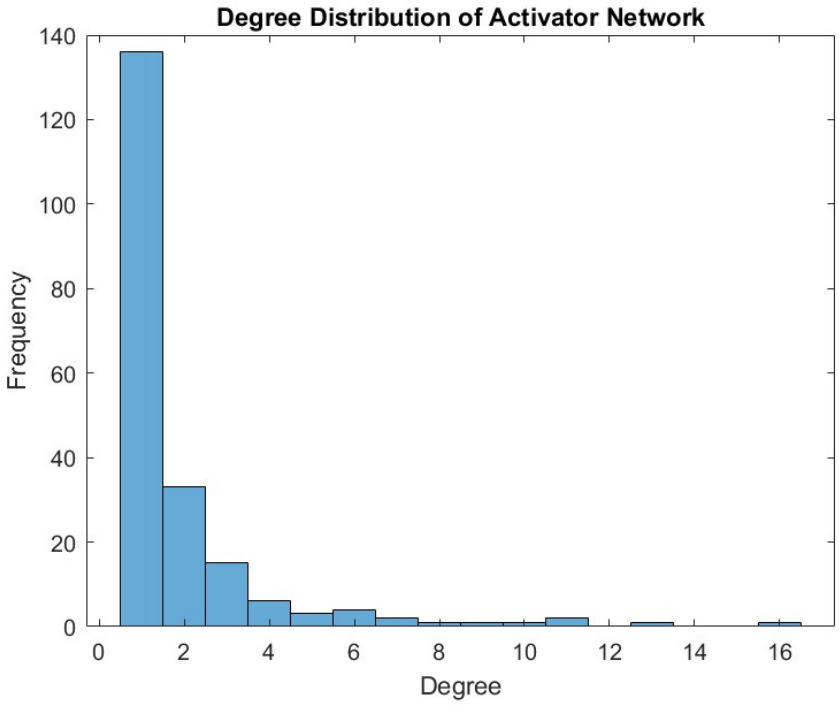
We find the transcription factors that contribute a higher fraction of activation in the network. Let $N(A)_i$ denote the number of activations by transcription factor i. The fraction of activation contributed by the transcription factor i to the network is given by:

$$f(A)_i = \frac{N(A)_i}{\sum_i N(A)_i}$$

The top 5 transcription factors with the highest activation fractions are found to be:

Rank	Transcription factor	Activation fraction
1	crp	0.16119
2	rpoE_rseABC	0.074627
3	yhdG_fis	0.074627
4	fnr	0.047761
5	nlpD_rpoS	0.041791

These transcription factors are then removed from the network. Next, only the activating edges are retained and the repressing and dual regulating edges are removed. This gives us the **activator network**. We then compute the degree distribution of this network, and the degree centrality, and closeness centrality of each of its nodes. We find the top 5 activators according to the centrality measures.



Rank	Activator	Degree centrality
1	fliAZY	16
2	rpoN	13
3	himA	11
4	rob	11
5	rpoH	10

Rank	Activator	Closeness centrality
1	fliAZY	0.00033908
2	rpoN	0.00032747
3	hns	0.0002725
4	rob	0.00026358
5	himA	0.00026175

Problem 2D

We find the transcription factors that contribute a higher fraction of repression in the network. Let $N(R)_i$ denote the number of repressions by transcription factor i. The fraction of repression contributed by the transcription factor i to the network is given by:

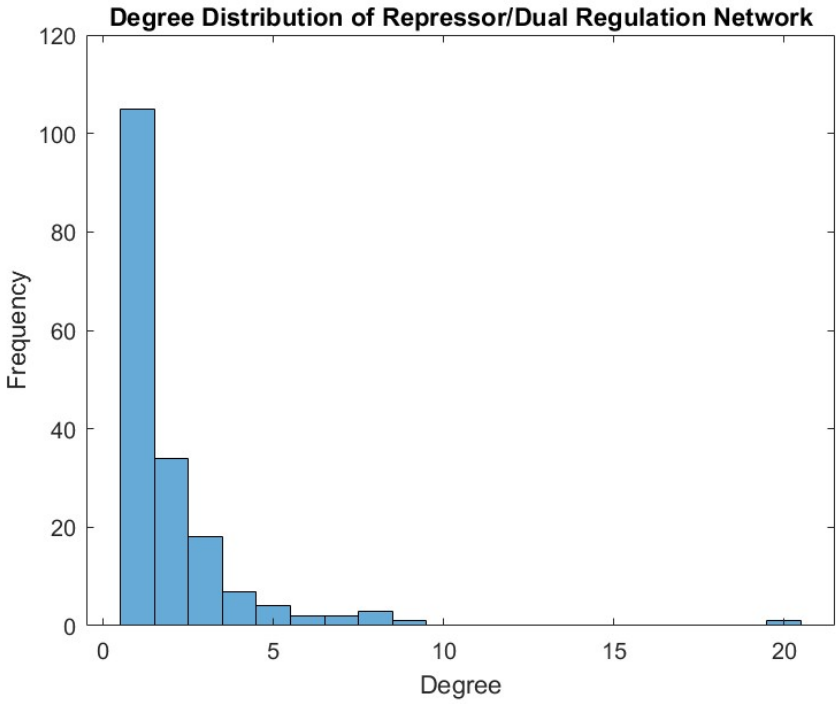
$$f(R)_i = \frac{N(R)_i}{\sum_i N(R)_i}$$

The top 5 transcription factors with the highest repression fractions are found to be:

Rank	Transcription factor	Repression fraction
1	purR	0.079439
2	arcA	0.074766
3	lexA_dinF	0.056075
4	fur	0.046729
5	himA	0.037383

These transcription factors are then removed from the network. Next, only the repressing and dual regulating edges are retained and the activating edges are removed. This gives us the **repressor/dual regulator network**. We then compute the degree distribution of this network, and the degree centrality, and closeness centrality of each of its nodes. We find the top 5 repressors/dual regulators according to the centrality measures.

Rank	Repressor/dual regulator	Degree centrality
1	crp	20
2	lrp	9
3	argR	8
4	fnr	8
5	tyrR	8



Rank	Repressor/dual regulator	Closeness centrality
1	crp	0.0005811
2	cytR	0.00022598
3	lrp	0.00022598
4	narL	0.00022598
5	argR	0.0001937