Simon Gibson 662005956
Frontiers of Network Science
October 16, 2023

Homework 1

1. Fungal Mesoscopic Response Functions

In the paper *Mesoscale analyses of fungal networks as an approach for quantifying phenotypic traits.*, the authors used the structural composition and mesocale observation of various fungal growths. In order to measure how fungal colonies contribute to their ecosystems, fungal growths were observed for 81 days at 3 day intervals under a multitude of conditions, including external attack from other species. At each snapshot in time, two networks are created, one structural and the other functional. Provided by the authors are numerous data sets, of which I chose day 31 of *Phanerochaete velutina: a foraging saprotrophic woodland fungus that forms reasonably dense networks.* The resulting network is calculated with nodes as hyphal tips, and edges as chords between cells. Each edge was asssigned a weight which (to my best understanding) is a measure of either the nutrient transport along said chord, or the ionic permeability of the hyphal tips.

1.1. Data.

The presented data came in a format which could not be used by either Gephi or NetworkX, so I have done a large amount of preprocessing, removing edge weights and converting to a simple edge adjacency-list format. To do this, I created a simple matlab script `mat_to_csv.m` to read in a matlab object file and convert it so csv, and then manually to adjacency list. As my second network is quite large, I chose a relatively small network of ≈1500 Nodes and ≈2300 Edges. Below are statistics calculated for each of the three networks, as well as the accompanying distributions and best-fit lines.

1.2. Discussion.

Due to the basis of the network being biologically structural, the source and two derived graphs are almost all a single connected component with very low degree variance. An unfortunate fact I did not discover until well into the process of analyzing the source network. Due to this, clustering coefficients were almost all incredibly low with even lower variance. Similarly, the connected component distributions are useless at best.

Additionally, while the source network had a few hubs (one of degree 21, 24, and 28), the two derived networks failed to extrapolate any.

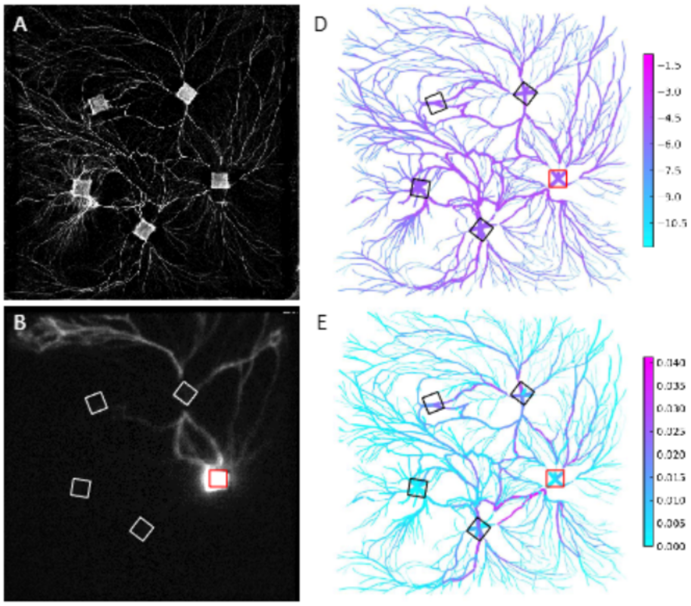
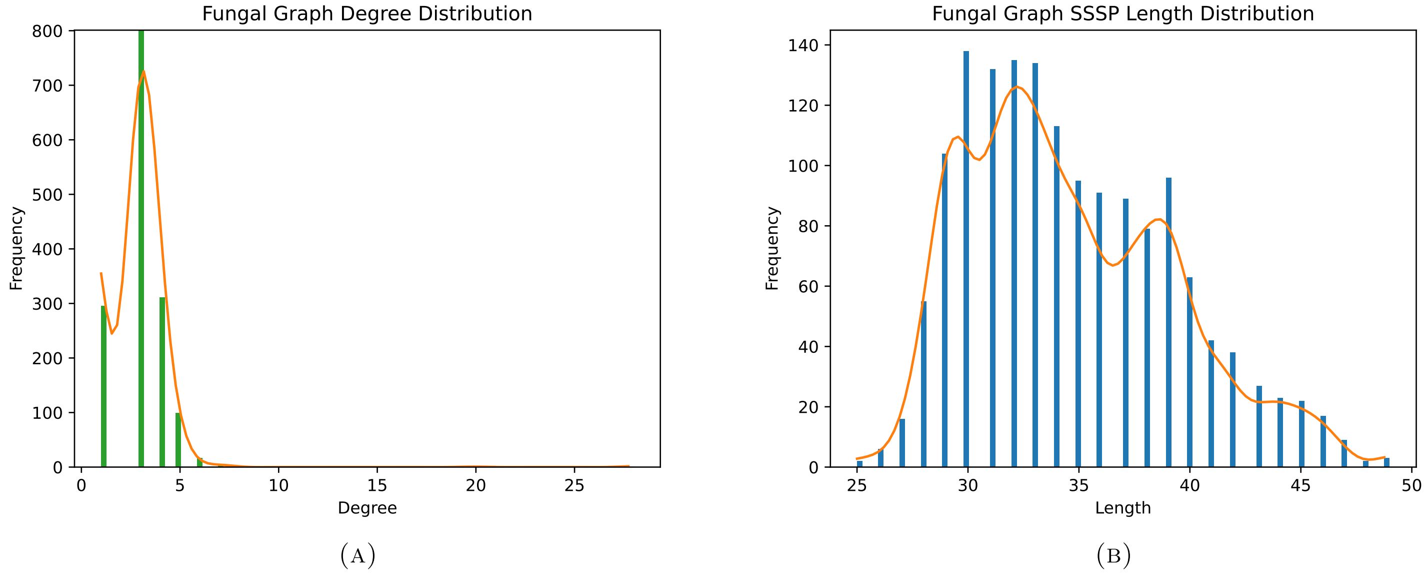


Figure 1. Phanerochaete velutina at 31 days, and corresponding structural (top-right) and functional (bottom-right) networks

1.3. **Source Network.** `Pv_M_I+R_U_N_31d_3.mat`

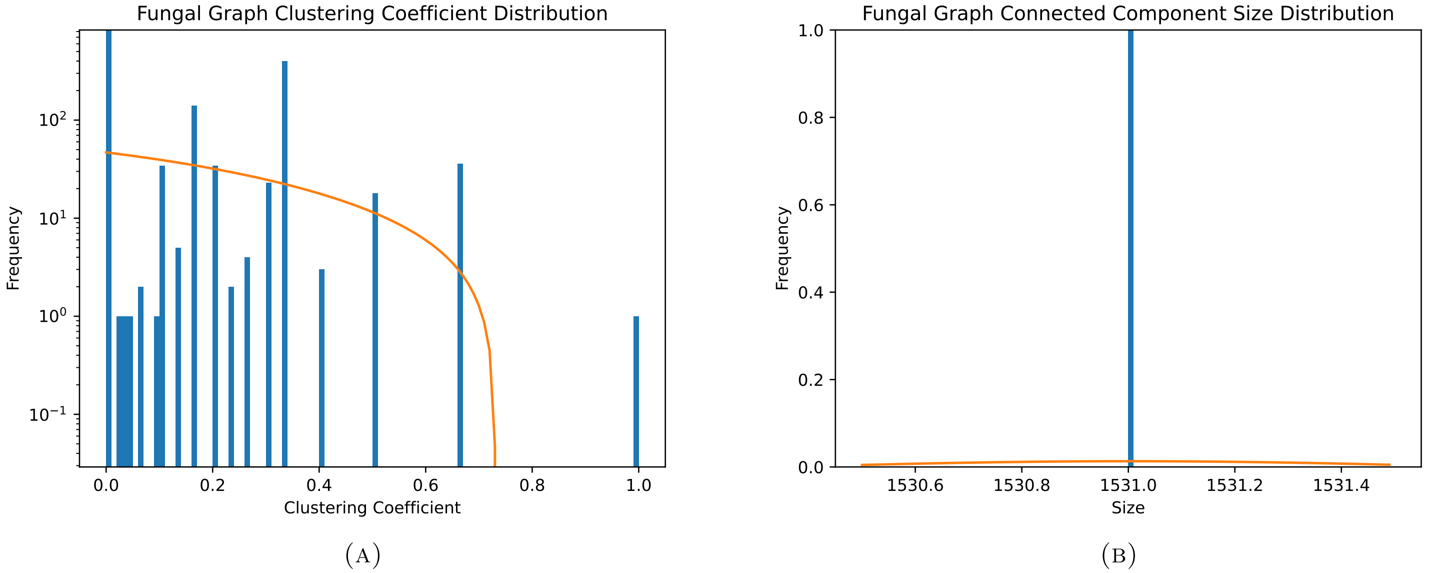
Fungal MRF							
Edges	2312	Eccentricity$_{\text{avg}}$	33.69	LSP$_{\text{avg}}$	33.6825	Conn. Com.$_{\text{max}}$	1531
Nodes	1531	Eccentricity$_{\text{var}}$	19.3824	LSP$_{\text{var}}$	21.6419	Conn. Com.$_{\text{var}}$	0.0
Radius	24	Degree$_{\text{var}}$	1.9897	Clust. Coeff.$_{\text{avg}}$	0.1373	Conn. Com.$_{\text{avg}}$	1531
Diameter	48	Degree$_{\text{avg}}$	3.0202	Clust. Coeff.$_{\text{var}}$	0.0296	# Conn. Com.	1



(A)

(B)

FIGURE 2. Degree and Longest Single Source Shortest Path Distributions



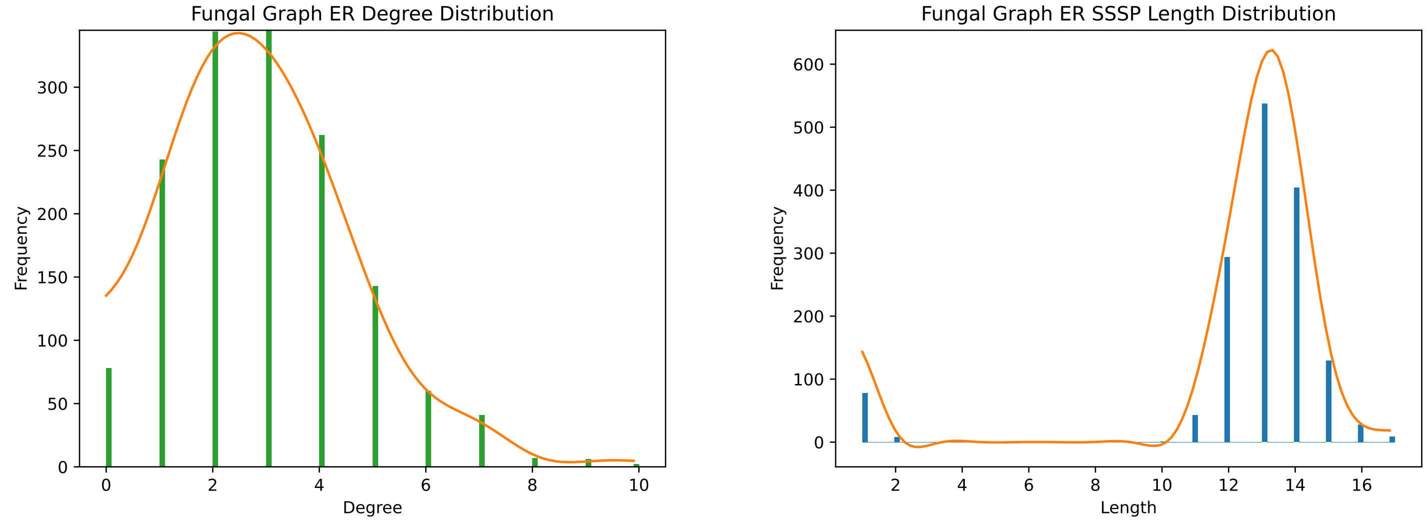
(A)

(B)

FIGURE 3. Cluster Coefficient and Connected Component Size Distributions

1.4. **Erdos-Reyni Random Network.** With edge probability $p = 1.972e - 3$

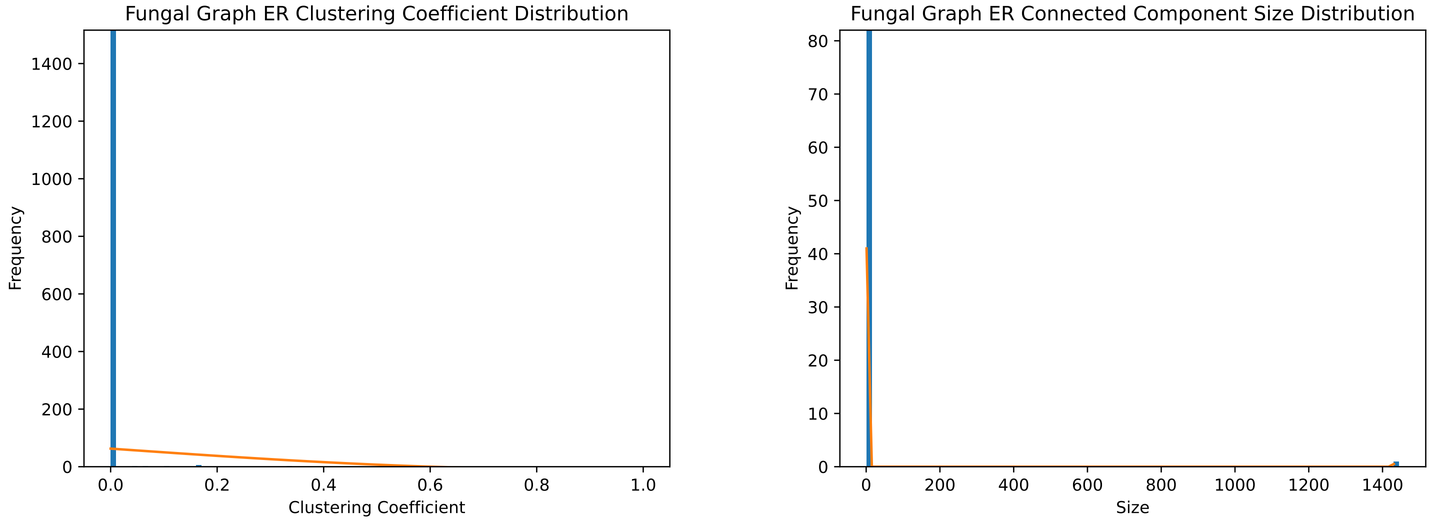
Fungal MRF- Erdos Reyni							
Edges	2253	$\text{Eccentricity}_{\text{avg}}$	11.5918	LSP_{avg}	12.5918	$\text{Conn. Com.}_{\text{max}}$	1445
Nodes	1531	$\text{Eccentricity}_{\text{var}}$	77.8881	LSP_{var}	8.9719	$\text{Conn. Com.}_{\text{var}}$	24817.8133
Radius	0	$\text{Degree}_{\text{var}}$	2.9602	$\text{Clust. Coeff.}_{\text{avg}}$	0.0020	$\text{Conn. Com.}_{\text{avg}}$	18.4458
Diameter	16	$\text{Degree}_{\text{avg}}$	2.9432	$\text{Clust. Coeff.}_{\text{var}}$	0.0009	# Conn. Com.	83



(A)

(B)

Figure 4. Degree and Longest Single Source Shortest Path Distributions



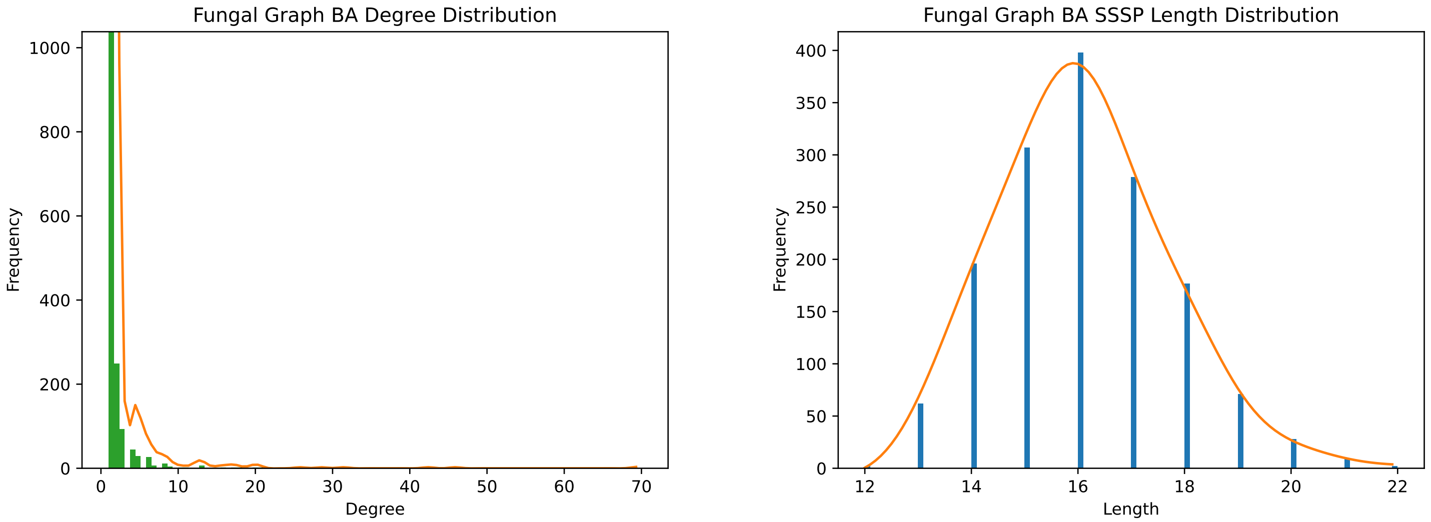
(A)

(B)

Figure 5. Cluster Coefficient and Connected Component Size Distributions

1.5. **Barabasi-Albert Random Network.** With average degree $d = \min(d(\text{Source Network}))$

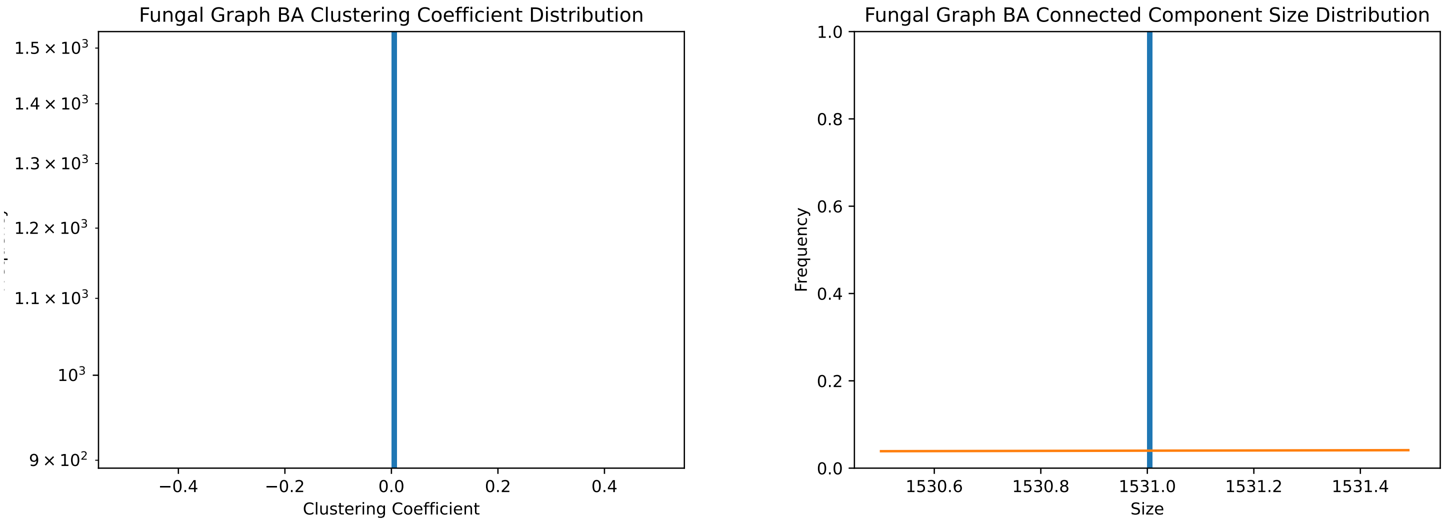
Fungal MRF - Barabasi Albert							
Edges	1530	$\text{Eccentricity}_{\text{avg}}$	11.7969	LSP_{avg}	12.7968	Conn. Com._{max}	1531
Nodes	1531	$\text{Eccentricity}_{\text{var}}$	16.04161	LSP_{var}	1.7895	Conn. Com._{var}	0.0
Radius	8	$\text{Degree}_{\text{var}}$	11.7962	Clust. $\text{Coeff.}_{\text{avg}}$	0.0	Conn. Com._{avg}	1531
Diameter	16	$\text{Degree}_{\text{avg}}$	1.9986	Clust. $\text{Coeff.}_{\text{var}}$	0.0	# Conn. Com.	1



(A)

(B)

Figure 6. Degree and Longest Single Source Shortest Path Distributions



(A)

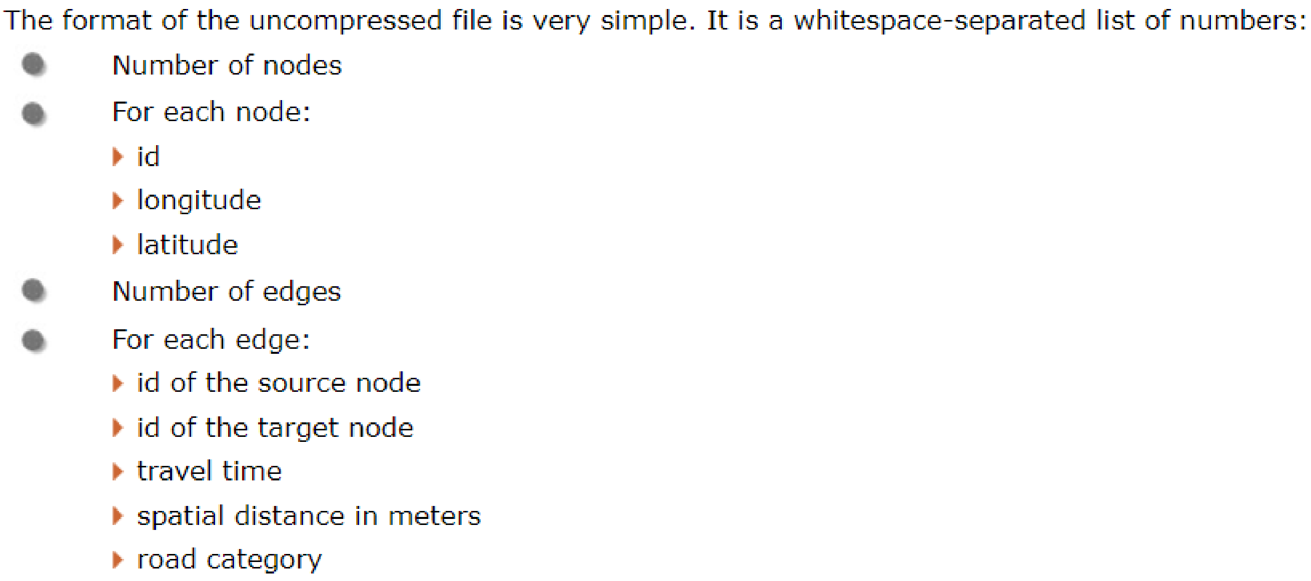
(B)

Figure 7. Cluster Coefficient and Connected Component Size Distributions

2. 9th DIMACS Implementation Challenge - District of Columbia

The Center for Discrete Mathematics and Theoretical Computer Science has hosted annual network based challenges for numerous years. For the 9th annual challenge, gathered from the U.S. Census Bureau TIGER/LINE road network files, the DIMACS challenge was to implement the fastest SSSP algorithm. Thankfully, (or not in the case of my computer processor), US states have great road infrastructure, with road counts in the 6-7 figure range. Of these available states/territories, District of Columbia has the smallest and most approachable network. As probably the most apt metaphor for a network, roads are considered edges and intersections as nodes.

2.1. **Data.** Similar to the Fungal MRF Data, the provided format was unusable by Gephi or NetworkX. The given format was



The format of the uncompressed file is very simple. It is a whitespace-separated list of numbers:

- Number of nodes
- For each node:
  - id
  - longitude
  - latitude
- Number of edges
- For each edge:
  - id of the source node
  - id of the target node
  - travel time
  - spatial distance in meters
  - road category

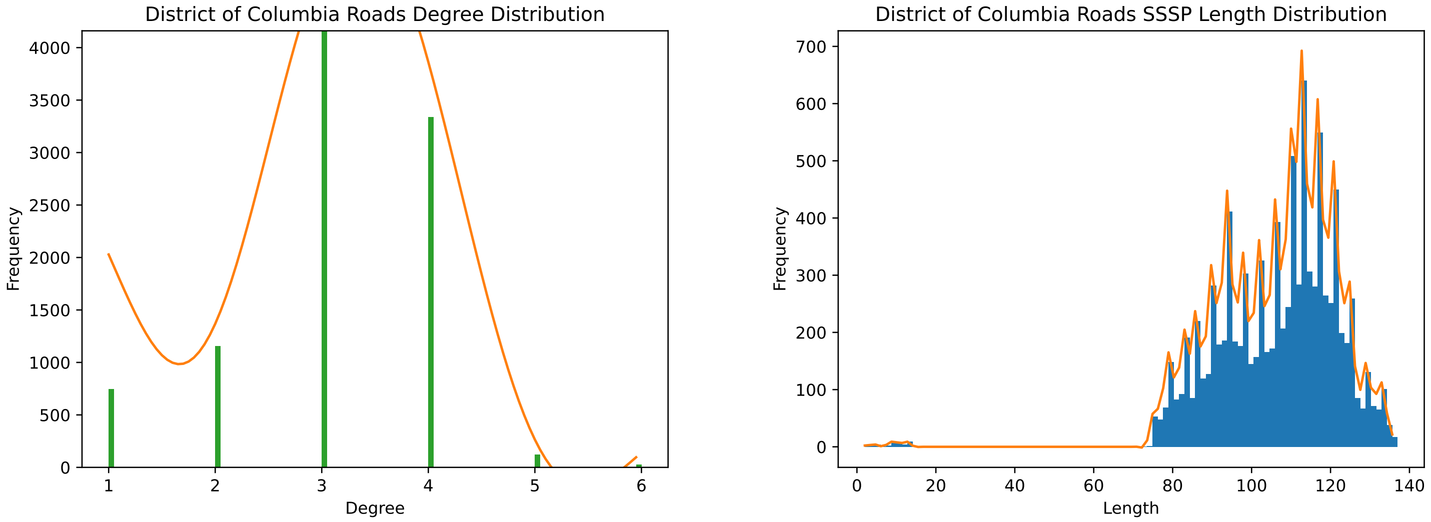
Figure 8. DC Road Network Format

To which I converted to the same format as the first network of a simple edge adjency list. To do this, I first used the DIMACS provided perl script `tiger2edimacs.pl` to convert to .gr format, and then manually to adjacency list. Additionally, this network came directed, with duplicates for each direction of the road. So despite the claimed edge count of 14,000, it is in reality 29,000. This network was much larger than the first, at ≈9,600 nodes, and ≈29,000 edges.

2.2. **Discussion.** Again similar to the first network, do the physical basis of the network, it is fully connected, and so the two derived graphs have few clusters if not one. The larger size of this network, however, allowed me to find numerous bugs and defects in my implementations. One such issue lies in the homework assignment, which states that the edge probability in the Erdos-Reyni graph should be the average degree of the source graph. With $\text{degree}_{\text{avg}} \approx 3$, this leads to a complete network of 91 million edges (an amount way beyond my laptops processing power). After reviewing notes, the probability is instead $p = \frac{\text{degree}_{\text{avg}}}{\text{no. nodes}}$. As it would be a traffic nightmare to have an intersection of roads above single digits, there are very few if any hubs in this network. If I could go choose networks again, I would first analyse the connected components and degree distributions before making a decision.

2.3. **DC Roads Source Network.** `DC.tmp`

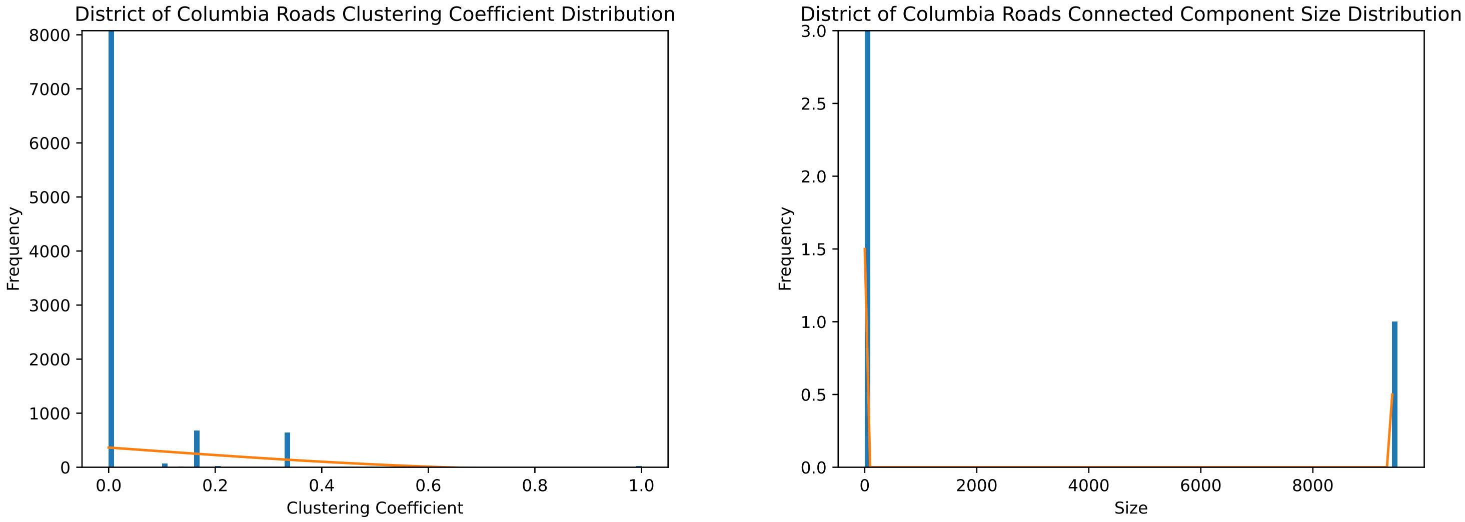
DC Roads							
Edges	14836	Eccentricity$_{\text{avg}}$	105.7712	LSP$_{\text{avg}}$	106.7712	Conn. Com.$_{\text{max}}$	9513
Nodes	9550	Eccentricity$_{\text{var}}$	N/A	LSP$_{\text{var}}$	229.3911	Conn. Com.$_{\text{var}}$	16924356.25
Radius	1	Degree$_{\text{var}}$	0.8484	Clust. Coeff.$_{\text{avg}}$	0.0390	Conn. Com.$_{\text{avg}}$	2387.5
Diameter	136	Degree$_{\text{avg}}$	3.1070	Clust. Coeff.$_{\text{var}}$	0.0111	# Conn. Com.	4



(A)

(B)

FIGURE 9. Degree and Longest Single Source Shortest Path Distributions



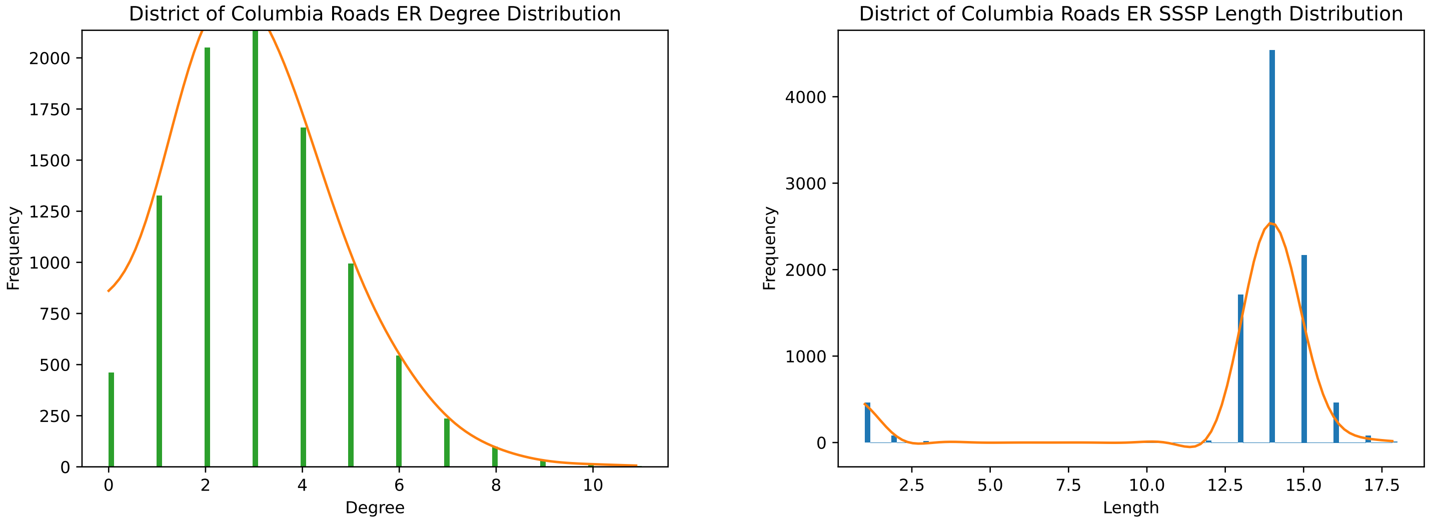
(A)

(B)

FIGURE 10. Cluster Coefficient and Connected Component Size Distributions

2.4. **Erdos-Reyni Random Network.** With edge probability $p = 3.253e-4$

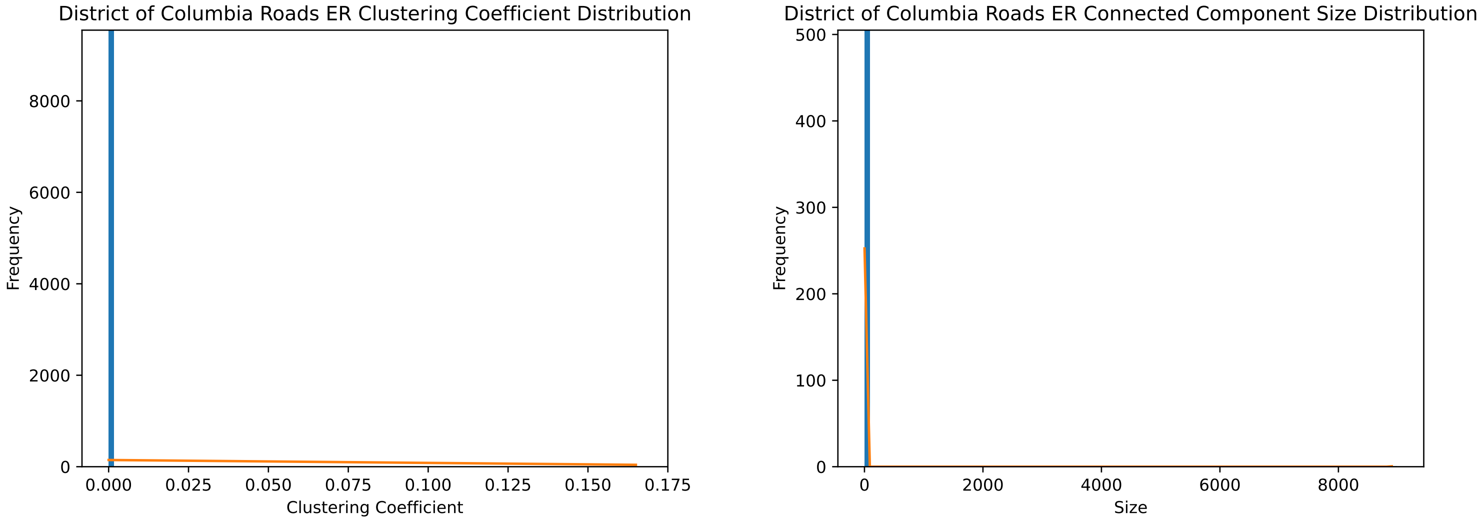
DC Roads - Erdos Reyni							
Edges	14790	$\text{Eccentricity}_{\text{avg}}$	12.4211	LSP_{avg}	13.4211	Conn. Com._{max}	8991
Nodes	9550	$\text{Eccentricity}_{\text{var}}$	N/A	LSP_{var}	9.9495	Conn. Com._{var}	159404.2053
Radius	0	$\text{Degree}_{\text{var}}$	3.1557	Clust. $\text{Coeff.}_{\text{avg}}$	3.1662e-5	Conn. Com._{avg}	18.8735
Diameter	17	$\text{Degree}_{\text{avg}}$	3.0973	Clust. $\text{Coeff.}_{\text{var}}$	4.0883e-6	# Conn. Com.	506



(A)

(B)

FIGURE 11. Degree and Longest Single Source Shortest Path Distributions



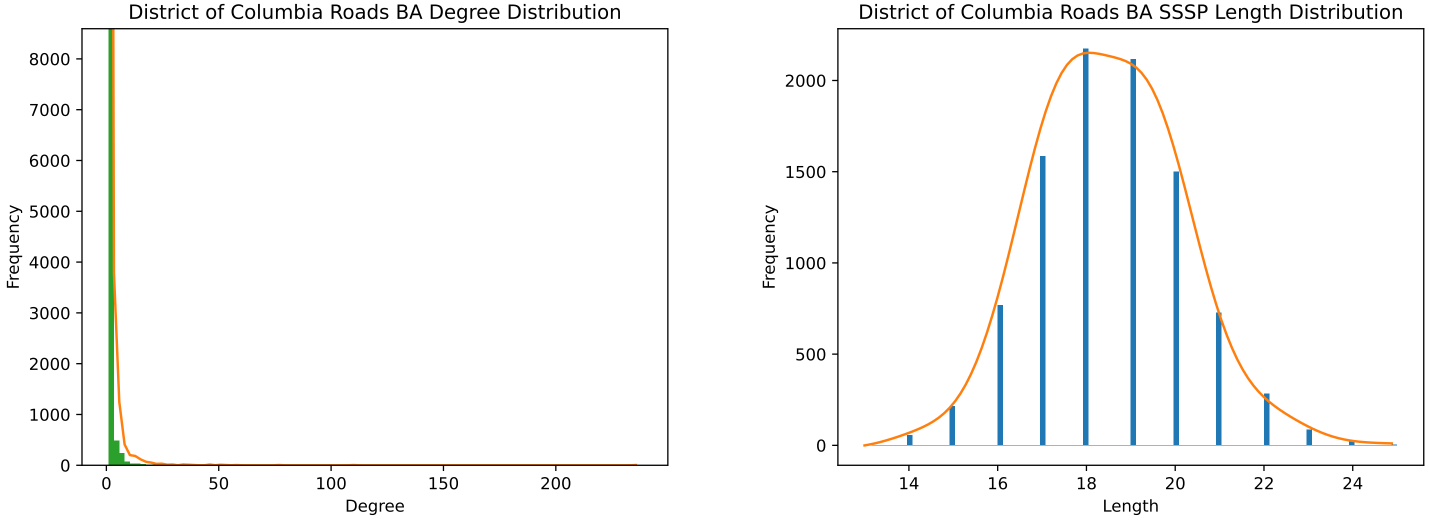
(A)

(B)

FIGURE 12. Cluster Coefficient and Connected Component Size Distributions

2.5. **Barabasi-Albert Random Network.** With average degree $d = \min(d(\text{Source Network}))$

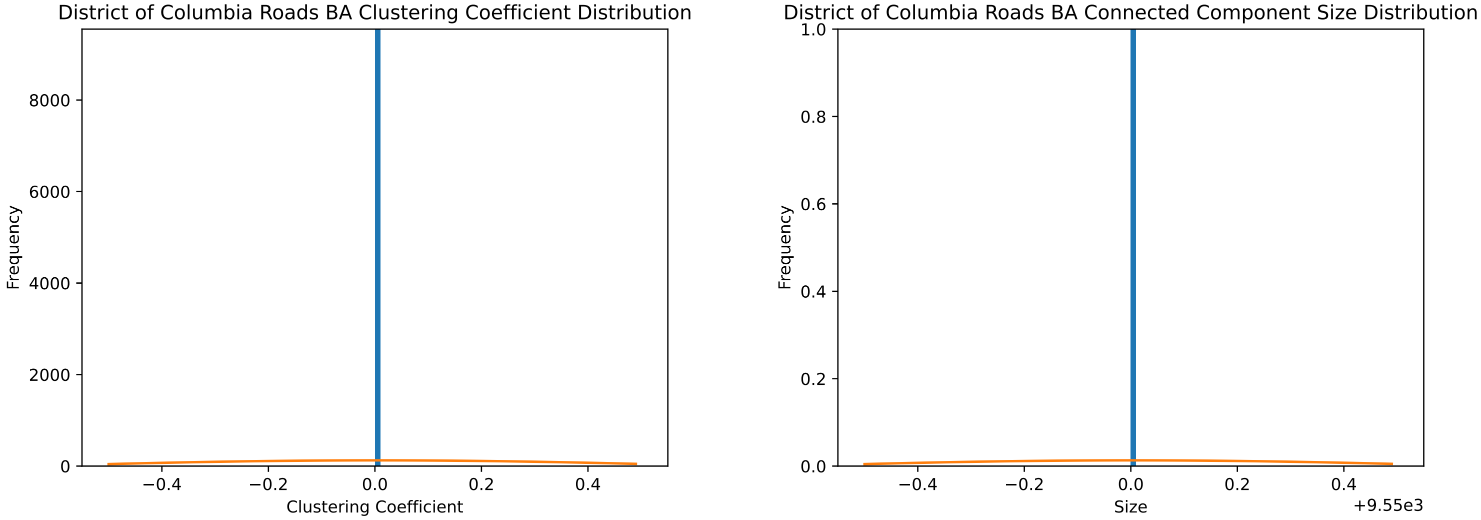
Fungal MRF							
Edges	9549	$\text{Eccentricity}_{\text{avg}}$	17.5312	LSP_{avg}	18.5312	Conn. Com._{max}	9550
Nodes	9550	$\text{Eccentricity}_{\text{var}}$	N/A	LSP_{var}	2.8739	Conn. Com._{var}	0.0
Radius	12	$\text{Degree}_{\text{var}}$	15.3803	Clust. $\text{Coeff.}_{\text{avg}}$	0.0	Conn. Com._{avg}	9550.0
Diameter	24	$\text{Degree}_{\text{avg}}$	1.9997	Clust. $\text{Coeff.}_{\text{var}}$	0.0	# Conn. Com.	1



(A)

(B)

Figure 13. Degree and Longest Single Source Shortest Path Distributions



(A)

(B)

Figure 14. Cluster Coefficient and Connected Component Size Distributions

3. Implementation

Originally, I intended to use Gephi for most if not the entire assignment. However, as I began to progress, I quickly reached numerous road-blocks, and instead switched to NetworkX, a very robust python library for network science. My script works on a single network at a time, where every statistic is calculated. In order to save on processing time while writing my script, it works on only a single network at a time, but can easily be modified for more. In order to analyze the derived Erdos-Reyni and Barabasi-Albert networks, changing the values of booleans `doBA, doER` to true will suffice. The expected input should be a whitespace delimited adjacency list, with each line as `node_label_1 node_label_2` designating one edge.

My biggest setback in implementing this assignment was not pertaining to the networks themselves, but massive difficulty in approximating a distribution from so few bins (as was the case in many of the distributions). Originally, I wished to show that the Erdos-Reyni and Barabsi-Albert networks followed the expected Powerlaw and Poisson degree distributions, but ran into many difficulties. The first implementation fit the probability density function each of a normal, powerlaw, and poisson distribution to the histogram, and then evaluated it on a linear space from minumum to maximum degree. However, due to poisson being discrete, and the other two continuous, many issues arose. Next I tried polynomial fits of many degrees, all to no avail, and then a random forest regressor, which overfit the data to the point of mimicking the histogram perfectly. As a compromise I finally landed on a nonlinear least squares with RBF kernel.

Another issue between my implementation and the assignment arose in Barabasi-Albert networks. As stated, the input parameter should be the minumum degree of the source network. However, I believe this was 1 in the case of both source networks. This resulted in both derived Barabasi-Albert networks becoming a tree, with no triangles, cycles, or interesting connected components.

3.1. Running.

In order to run my script, the dependencies `numpy, scikit-learn, pandas, matplotlib`, and `networkX` are required. The variables `path` and `title` decide the input adjacency list, and output formatting, respectively. Python 3.6 or higher is required. All necessary scripts, network sources, and modified networks are provided in `Homework1.zip`.