

# Assignment 1

COMP 596: Network Science

Due on September 20th 2020

Can be submitted individually or in groups of two (recommended).

1. Choose 3 of the datasets in the Barabasi book, compute and plot [65%]: consider simple graphs, i.e. remove self-loops, multi-edges, and directions (make them symmetric).
  - (a) the degree distribution (fit a line and find the slope),  
useful functions: `numpy.loadtxt()`, `scipy.sparse.csc_matrix()`, binning and log log scale, `numpy.polyfit()`
  - (b) clustering coefficient distribution (compute the average as well),  
if not computationally feasible for some of the graphs, don't report for that specific graph and specify the reason; same is true for other parts
  - (c) shortest paths distribution (compute the average as well),  
you can use `scipy.sparse.csgraph` functions, as well as sampling only fraction of nodes to estimate the dist of all pair shortest paths for larger graphs, or drop them similar to above
  - (d) number of connected components, the portion of nodes that are in the GCC (giant/largest connected component)  
useful functions: `csgraph.connected_components()`
  - (e) eigenvalue distribution (compute the spectral gap),  
useful functions: `sparse.linalg.eigs()`
  - (f) degree correlations (plot as scatter  $d_i$  vs  $d_j$ , also report the overall correlation).  
plot degree of source vs degree of destination, axes would be 0 to max degree in the graph and you have a point in the scatter plot if the corresponding degree values are connected by any edge use counting, binning or plot edges with low intensity to capture regions with high density
  - (g) degree-clustering coefficient relation (plot as scatter  $d_i$  vs  $c_i$ )  
scatter plot with axes being local cc and degree. Plot each node to show how degree correlates with the cc of nodes. Similar to the plot above, capture nodes plotted over each other
2. Report the computational complexity for a-g [5%]
3. Implement the AB model, compute the same distributions for the synthetic networks of similar size as 1, generate one sample per real network. [30%] same number of nodes, estimate edges added in each iteration based on total number of edges

bonus Tweak the AB model and report the effects on observed patterns [5%]

If using python, you can use sparse matrices in scipy, but we are not going to use Networkx (or similar network analysis packages) for this assignment (implement from scratch). Submit the report in pdf and code as separate attachments, through Mycourses.