

# Graph Theory Homework 1

Student name and ID

Istanbul Technical University, Turkey

(1) Questions 1 and 3: All codes should be written either in C/C++ or Python, and integrated as new subpackages into SNAP library <https://snap.stanford.edu/>. Code integration means you use SNAP library functions/tools to develop your code and define a new program that can be executed using a command line in a terminal as specified by the programmer (in a similar way to existing SNAP functions).

(2) Questions 2: To implement hypergraph centrality, feel free to use Python or C/C++ without using SNAP library. The ‘HypergraphCentrality’ program can be independent. However, make sure you properly explain in the Readme.txt file how it works, how to test it, and which paper was used to implement it. **Any code plagiarism will lead to assigning ‘0’ mark to this homework.**

(3) Use this Latex file to write and submit your final report (both .tex and PDF files).

(4) Include a ‘Readme.txt’ file for ALL your codes to explain to the user how to test your program on new data (specify inputs, outputs, how the program works, etc).

## 1 Profiling a brain network using different centrality measures [45%]

A functional brain graph  $G$  encoded in an  $N \times N$  matrix, where  $N = 116$  denotes the number of anatomical regions in the brain. The graph adjacency matrix  $\mathbf{C}$  is stored in ‘C.mat’ file (**Fig. 2**).

**Step 1: network binarization** [5%]. Noting that  $\mathbf{C}$  has negative edges, **binarize**  $\mathbf{C}$  at  $\rho = 0$ . This will produce a *binary matrix preserving only edges with positive weights*. Display the produced **binarized** matrix in your report.

**Step 2: node centralities** [10%].

**2.1 Using the binarized  $\mathbf{C}$  matrix**, generate an overlapping barplot (see example in **Fig. 1**) of centrality values of all 116 nodes in  $G$  using the following measures (these can be found in SNAP <https://snap.stanford.edu/>):

- (1) degree centrality
- (2) eigenvector centrality
- (3) closeness centrality

**2.2** Discuss the plot. What do you notice? Any particular vital nodes? What if those nodes are lesioned (or removed), how would that affect brain function?

**Step 3: centrality distributions** [10%].

**3.1** Generate a plot where you display the overlapping distributions of the three computed node centralities of  $\mathbf{C}$ .

**3.2** Discuss the plot. What conclusions can you derive about the network topology and integrity?

**Step 4: brain connectome percolation** [20%].

**4.1** Integrate a new package into SNAP called ‘findPercolationThreshold’, where you input a positive weighted adjacency matrix and it automatically outputs its percolation threshold  $p_c$ .

**4.2** Detail the fundamental steps of your algorithm and generate a similar plot to Figure 6.4 in FBNA textbook using the strategy described in the textbook as well as at minute 4:50 in Graph Theory Blink 3.3<sup>a</sup>.

**Remark 1:** In Figure 6.4, for each graph, they gradually add edges in decreasing order of weight and plot the size of the largest weakly connected component. To extract the largest weakly connected component and compute its size, you can use `GetMxWcc` and `GetMxWccSz` in SNAP library. **As for the percolation threshold, you should define it when the size of the largest weakly connected component becomes about 95% of the total graph size (i.e., the largest weakly connected component contains 95% of nodes in the graph). Feel free to generate random binary adjacency matrices and check how your code works on those.**

**Remark 2:** (1) The code for each subquestion (e.g., Q1.1) should be submitted along with the written report, otherwise the question will not be marked if it required coding up. (2) Please also include screenshots of your new code in the written report. (3) Include a ‘.txt’ file to instruct the user on how to use and test your code.

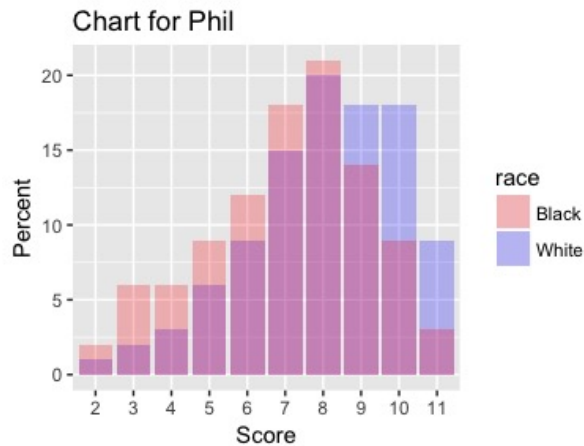


Fig. 1: A random example of an overlapping barplot..

<sup>a</sup> <https://www.youtube.com/watch?v=TDc4YakqUZs&t=8s>

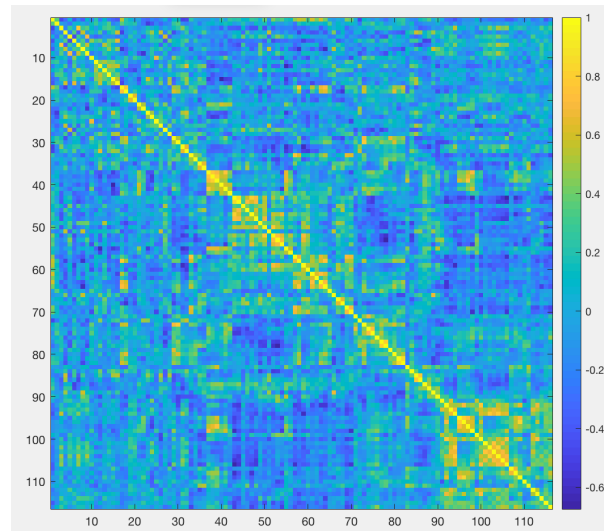


Fig. 2: *Adjacency matrix of the functional brain connectome  $C_{..}$ .*

## 2 Hypergraph centrality [30%]

Let  $\mathbf{H}$  denote the hypergraph incidence matrix stored in ‘H.mat’ file (**Fig. 3**).

**Step 1: Finding a good paper.** Based on your research, find a good paper explaining how to compute centrality measures on hypergraphs. Reference ([ ]) the paper and explain how the centrality measure is computed (include the mathematical formulation).

**Step 2: Coding up hypergraph centrality.** Code up the proposed hypergraph centrality and integrate your module entitled ‘hypergraphCentrality’ into SNAP. Your code should input an incidence matrix  $\mathbf{H}$  and output the centrality measures.

**Step 3: Hypergraph centrality profile.**

3.1 Plot the centrality distribution for hypergraph  $\mathbf{H}$ .

3.2 What conclusions can you derive about the hypergraph topology?

## 3 Profile a SNAP graph of your choice [25%]

**Step 1: Network selection and presentation** [5%]. Select a network (i.e., graph) from the SNAP pool available at <https://snap.stanford.edu/data/index.html>.

1.1 Explain in detail what the graph represents.

1.2 Randomly remove 40% of the nodes in the network and use the thresholded one to implement the following steps 2 and 3.

**Step 2: Network profiling using centralities** [10%].

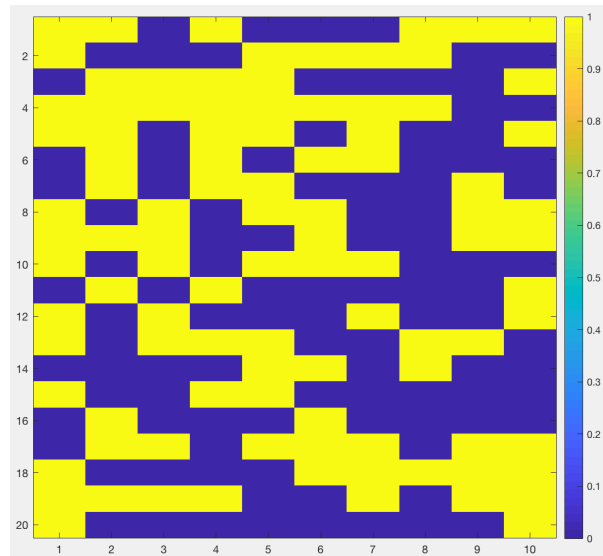


Fig. 3: *Hypergraph incidence matrix  $\mathbf{H}$ .*

**2.1** Generate a plot where you display the overlapping centrality distributions of the selected network using the 3 centrality measures (degree, eigenvector, and closeness).

**2.2** Discuss the plot. What conclusions can you derive about the network topology and integrity?

**Step 3: Connected components and resilience to attacks [10%].**

**3.1** Plot the sizes of strongly connected components (*strongly* if the network is directed otherwise the sizes of connected components in an undirected network) against the number of removed most central nodes in the network. To do so, you can sort nodes by their centralities in a decreasing order and remove the top 5% most central nodes, then 10%, until you reach 60% with a step size of 5%.

Generate superimposed (or overlapping scatter plots) using:

- (1) degree centrality
- (2) eigencentrality
- (3) betweenness centrality

**3.2** What conclusions can you derive about your network topology and its resilience to attacks targeting its central nodes?

You can use SNAP modules at <https://snap.stanford.edu/snappy/doc/reference/cncom.html>