

Homework #0

This homework is easy and is meant to get you started with network analysis. The homework will not be graded, but if you find it hard, then this course might not be for you. Any questions and comments regarding the homework should be directed to [Piazza](#).

Submission details

This homework is due on **March 5th** at 9:00pm, while late days expire on **March 8th** at 12:00pm. The homework must be submitted through (1) [Gradescope](#) (entry code **NM8ZRM**) and (2) [eUcilnica](#). (1) Submission to [Gradescope](#) should include answers to all questions, each on a separate page, which may also demand pseudocode, proofs, tables, plots, diagrams and other. (2) Submission to [eUcilnica](#) should include at least this cover sheet with signed honor code and all the programming code used to complete the exercises. The homework is considered submitted only when *both* parts have been submitted. Failing to include this honor code in the submission will result in **10% deduction**. Failing to submit all the developed code will result in **50% deduction**.

Honor code

Students are strongly encouraged to discuss the homework with other classmates and form study groups. Yet, each student must then solve the homework by her/himself without the help of others and should be able to redo the homework at a later time. In other words, students are encouraged to collaborate, but should not copy from one another. Referring to any solutions obtained from classmates, course books, previous years, found online or other material, is considered an honor code violation. Also, stating any part of the solutions in class or on [Piazza](#) is considered an honor code violation. Finally, failing to name the correct study group members, or filling out the wrong date or time of the submission, is also considered an honor code violation. Honor code violation will not be tolerated! Any student violating the honor code will be reported to **faculty disciplinary committee** and vice dean for education.

Name & SID: _____

Study group: _____

Date & time: _____

I acknowledge and accept the honor code.

Signature: _____

1 Network software

Throughout the course you will be analyzing networks. This can be done using either some programming library or software tool. As you will be required to analyze large networks and also implement some algorithms on your own, you should first select your network library. The most popular network libraries include [NetworkX](#), [igraph](#), [Snap.py](#), [SNAP](#), [graph-tool](#), [JGraphT](#), [JUNG](#) etc. For most of these libraries you will be able to find documentation, tutorials, code examples, discussion groups and other online. It is strongly recommended that you select a library for the programming language you are most comfortable with.

At least for smaller networks, it is often beneficial to be able to get a quick insight into the network structure just by visualizing it. For this purpose, you should also select some network software. Most popular network software include [visone](#), [Gephi](#), [Pajek](#), [Graphviz](#) and [UCINET](#).

Finally, you are given the Zachary karate club network [[Zac77](#)] in [edge list](#), [Pajek](#) and [LNA](#) formats ([Figure 1](#)). This is a simple undirected network. Using your programming library, load the network and compute the number of nodes and edges. Using your software tool, try to neatly visualize the network.

2 Network collection

Within the course you will be analyzing various real-world networks. Prior to any analysis, one must first collect a network and clean it. Contrary to what you might expect, often more time is spent on preparing the network than on analyzing it afterwards.

2.1 Your own network

Map out a network on your own. Try to be creative and collect a *large* network that you find interesting or you have not seen before. Compute the number of nodes and edges, and the average node degree in your network using either your network library or software tool.

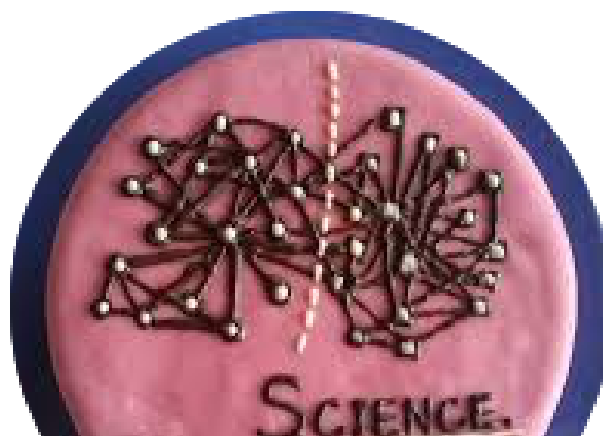


Figure 1: Zachary karate club network cake

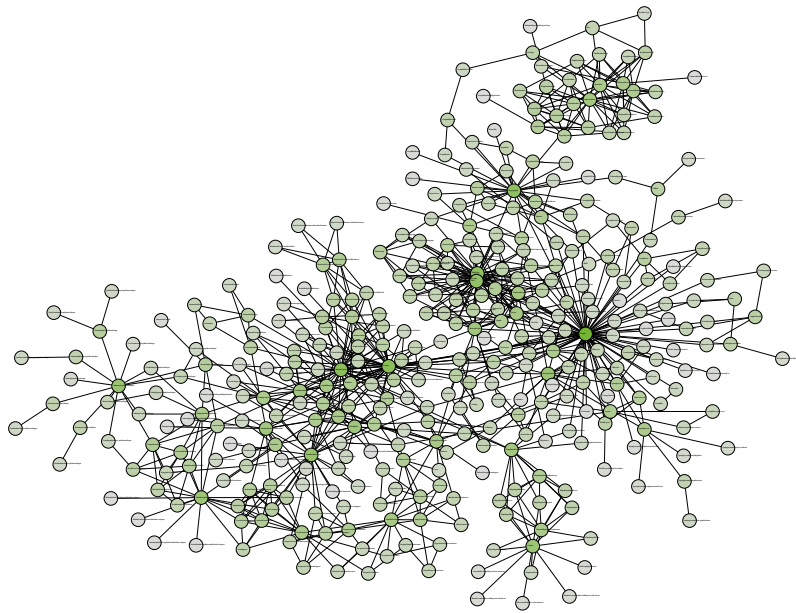


Figure 2: JUNG software network with node colors corresponding to PageRank

2.2 Synthetic graphs

Any network library or tool surely provides a way to generate the simplest possible synthetic graph called the Erdős-Rényi random graph [ER59]. Create an instance of the Erdős-Rényi random graph by setting any required parameters thus to best fit your network. Compute the number of nodes and edges, and the average node degree in the generated random graph.

3 Network analysis

Commonly not all nodes in a network are equally important (Figure 2). PageRank [BP98] is probably the most well known measure of node importance and it is therefore included in almost any network library. PageRank is the algorithm that put Google in front of other search engines more than two decades ago.

3.1 PageRank algorithm

Compute the PageRank score of all nodes in your network. Which nodes are most important according to PageRank and how would you interpret these results? Is the difference between the first and the tenth node substantial or are all first ten nodes somewhat equally important?

3.2 Networks vs graphs

Repeat the analysis for an Erdős-Rényi random graph with the same parameters as before. How do the PageRank scores compare to those obtained for your network? Is the difference between the first and the tenth node substantial or are all first ten nodes somewhat equally important?

References

- [BP98] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Comput. Networks ISDN*, 30(1-7):107–117, 1998.
- [ER59] P. Erdős and A. Rényi. On random graphs I. *Publ. Math. Debrecen*, 6:290–297, 1959.
- [Zac77] Wayne W. Zachary. An information flow model for conflict and fission in small groups. *J. Anthropol. Res.*, 33(4):452–473, 1977.