Complex networks in biology
Dynamical modelling of living systems 7.5hp

Lucas Hedström* and Ludvig Lizana†
IceLab, Umeå University

INTRODUCTION

In the preceding labs, you made molecular models of protein production and epigenetic gene regulation. You studied these systems in isolation to reach a mechanistic understanding of how they work. But in general, these systems are connected. Translation does not only occur with one mRNA sequence, but with various ones. What happens if the number of available amino acids fluctuate? Is this loosely connected to the already considered protein production? If the epigenetic landscape becomes methylated, the number of available methyl groups might reduce, and push the system towards the other steady state. Furthermore, this could have an impact on the aforementioned translation.

Simply put, cellular systems are connected in complex ways. So, to understand how cells function, we should consider them as being composed of many interconnected parts that form complex networks. In this lab, you will learn network tools and use them to analyze real data such as metabolic networks, protein-protein interaction networks, and gene regulatory networks. You will look at certain network parameters, compare them to random unstructured networks, and see how stable they are to perturbations.

Note: We use the word graph and network interchangeably.

Lab report

You will write a lab report that, without including figure captions, should consist of around of 1000–3000 words (roughly 3–5 pages). You are allowed one figure panel per task, but each figure panel can consist of more than one figure. Ensure that your plots are well-formatted in a vector format with clear axis labels and legends and proper font size.

Data

In this lab, you will work with three data files that you may download from Canvas. These are

- `yeast_gene_net.txt` — YeastNet v3 [1]. This is a network of functionally coupled genes, based on the gene's of baker's yeast.
- `ecoli_metabolic_net.txt` — partial metabolic network in E.coli from EcoCyc [2]. This network originally consisted of metabolite → reaction → metabolite. However, we pruned the reaction, producing a network where an edge denotes metabolites that are indirectly connected in a reaction.
- `human_kidney_protein_net.txt` — protein-protein interaction network in human kidney cells. This network describes how proteins are connected to form other, larger proteins.

All networks are unweighted and undirected. Unweighted means that there is no 'cost' going from one node to another via an edge. Undirected means that you can go in both directions on any edge.

These files are in the edgelist format, this means that each row represents an edge between the two nodes n_i and n_j as `ni,nj`[1]. In this lab, we strongly encourage you to use some network or graph package available for your programming language. *E.g.*, in MATLAB you can use the toolbox `Graph`, and Python has the module `networkX` [3, 4]. Most languages has an equivalent package that will do most of the work for you. Julia has `Graphs.jl`, Ruby has `GraphQL`, etc. If you are using a language such as C or C++, you can probably figure it out for yourself, but it's gonna be more hassle than what it's worth.

TASKS

Task 1: Determining the overall structure

Here you will calculate standard metrics that are often used to categorize networks. In subsequent exercises, you will compare these metrics to between the networks and their random counterparts.

Goal(s)

- Load the and calculate for each of the networks
  (i) The degree of each node.
  (ii) The local clustering coefficients (*i.e.* the clustering coefficient of each node).

[1] *E.g.*, the row `0,1` means that nodes 0 and 1 are connected.

(iii) The shortest path between all edge pairs.

- A metric that says alot about the type of network we have is the degree distribution $P(k)$, which is simply the probability that a node has a degree k. Calculate this quantity and plot this in a graph for all three networks. By reading up online, identify which networks are random and which are scale free. Discuss the ramifications this has on the networks.

- Plot histograms of the local clustering coefficient and shortest paths for all three networks. As in the before exercise, read up on the quantities, compare and discuss the differences between the networks.

Task 2: Comparison against random networks

Just looking at a single sample network might not be that insightful, especially considering that there is a lot of variability that can change the results. In this task we will tackle this by comparing the networks to a large-ish set of random networks.

We will generate Erdős-Réyni (ER) networks. If you're using a premade networks package you will probably find a function that generates these. If not, the algorithm to do this is very simple. Given a number of nodes N and a edge probability p the algorithm is as follows;

(i) Create a network with N nodes.

(ii) For all nodes n_i connect n_i to $n_j \neq n_i$ with a probability p.

If you're writing your own code to generate these networks, check that it follows the correct binomial distribution

$$P(k) = \frac{\langle k \rangle^k e^{-\langle k \rangle}}{k!},$$

before continuing.

Note: For some of these metrics, such as the average shortest path length and the network diameter (which you will be looking at task 3), you might get errors that the network is not fully-connected. This leads to some shortest paths being infinite and the network diameter not well-defined. To fix this, you should only look at the largest component of the graph, i.e. the connected subgraph that has most nodes, and ignore all other subgraphs.

Goal(s)

- Show theoretically that the average degree $\langle k \rangle = (N-1)p$.

- For each network, generate a good sample (20+) of ER networks which has the same average degree as the studied network. For each sample, compute the average clustering and average shortest path lengths. Plot these in histograms, with the value from the original networks clearly marked. What do you observe?

Task 3: Robustness in biology

A well-studied metric in network science is robustness. This metric aims to describe how well a network responds to perturbations. A good example is by thinking about a network of friends — if one friend is removed from a friend group — will the people left still be friends? How robust, *i.e.* how solid are the friendship contacts within the network? Is there just one person who keeps everyone together?

This metric is also applicable to biological networks. If a protein is removed from the cell, how will this cascade down to effects further down? Can other important proteins still be created? Is the protein just an auxiliary part that has no connections to other factors?

Goal(s)

- Explain what the network diameter is (it's very closely related to one of the metrics you've already looked at) and shortly discuss what a small/large diameter implies.

- For each of the networks, randomly remove a fraction f of the nodes and for each fraction calculate the diameter of the largest component. (Make sure you average over a few different removals since they will all be different). Plot the diameter for each network for f between 0 and 0.25.

- Do the same plot as the last task, however instead of randomly removing a fraction f instead remove the top fraction f of nodes with the highest degree. Compare the two removal methods. What do you observe? Discuss why this occurs and the implications. If you can, try to connect this discussion to the earlier figures.

* lucas.hedstrom@umu.se
† ludvig.lizana@umu.se

[1] H. Kim, J. Shin, E. Kim, H. Kim, S. Hwang, J. E. Shim, and I. Lee, Yeastnet v3: a public database of data-specific and integrated functional gene networks for saccharomyces cerevisiae, Nucleic acids research **42**, D731 (2014).

[2] I. M. Keseler, S. Gama-Castro, A. Mackie, R. Billington, C. Bonavides-Martínez, R. Caspi, A. Kothari, M. Krummenacker, P. E. Midford, L. Muñiz-Rascado, *et al.*, The ecocyc database in 2021, Frontiers in microbiology , 2098 (2021).

[3] Mathworks, Graph and Network Algorithms (2022), accessed 2022-08-09.
[4] NetworkX, NetworkX (2022), accessed 2022-08-09.