Graph Theory and Network Features

Practical Lab: Analyzing Gene Regulatory Networks

Sergio Peignier

1 Practical Objectives

In this practical lab, students will explore the following objectives:

- Load a real-world Gene Regulatory Network (GRN) from benchmark data.
- Quantify **local** and **global** topological characteristics.
- Identify structural patterns such as **network motifs** and **communities**.
- Benchmark the biological network against **Erdős-Rényi**, **Watts-Strogatz**, and **Barabási-Albert** random models.

2 Local & Global Features

Task

Load your assigned GRN dataset using NetworkX and analyze its basic metrics.

1. What are the size (nodes/edges) and density of the network? How do in-degree and out-degree distributions differ?
2. Compute the global clustering coefficient and average path length (if computing the average path length is too long, consider programming a function that estimates the average path for a subsample of nodes). Do these values suggest a small-world property?
3. Identify the top 5 "hub" nodes based on degree or centrality. What might high-degree transcription factors represent biologically?
4. Plot the in-degree and out-degree distributions of the network on a log-log scale. How do the two distributions differ in range and shape, and what do they indicate about the presence of master regulators and the scale-free nature of the GRN?

3 Motifs and Subgraphs

Task

Investigate local connection patterns and recurring subgraphs.

1. Run a triadic census on the directed GRN. Identify which non-empty triad types (excluding the dominant `003` empty triad) appear most frequently and discuss what these patterns suggest about the local architecture of the network.

2. Quantify how these motif frequencies compare against random expectations by writing a Python function that:

   (a) Takes the directed graph and generates a degree-preserving randomized version using `nx.algorithms.swap.directed_edge_swap` or `nx.directed_configuration_model`.

   (b) Computes the triadic census for the randomized network.

   (c) Repeats this process N times (e.g., $N = 50$) to calculate the Z-score for each triad type using the mean and standard deviation from the randomized runs:

$$Z_i = \frac{N_i^{\text{real}} - \langle N_i^{\text{rand}} \rangle}{\text{std}(N_i^{\text{rand}})}$$

3. Based on your Z-scores, which specific motifs are significantly overrepresented ($Z > 2$) or underrepresented ($Z < -2$) in the real GRN, and what might this reveal about biological selection pressures?

4 Communities

Task

Detect functional modules.

1. Apply a community detection algorithm, e.g., Louvain or Modularity maximization (`greedy_modularity_communities`).

2. How can you determine if the most important communities correspond to known functional pathways? Take a look at the `enrichr` function from the `gseapy` library.

5 Random Comparison

Task

Benchmark against random graph models.

1. Generate corresponding Erdős-Rényi, Watts-Strogatz, and Barabási-Albert models matching your network's basic dimensions.

2. Discuss which random model best captures the features of the real GRN and explain why.