**Università degli Studi di Padova**

Computer Engineering

# Learning From Networks Project Report

Human and mouse similarities in gene regulatory networks

Alessio Cocco - 2087635
Andrea Valentinuzzi - 2090451
Giovanni Brejc - 2096046

Academic Year 2023-2024

## Abstract

The scientific literature shows that GRNs dynamically change across conditions and a comprehensive comparison between human and mouse GRNs is therefore essential to understanding and curing diseases.
This report aims at understanding the similarities and differences between human and mouse gene regulatory networks by comparing network node metrics such as degree centrality, clustering coefficient, betweenness centrality, eigenvector centrality and counting triangles.

## Code

Datasets used are available at the following links:

- Human gene regulatory network: https://networkrepository.com/bio-human-gene1.php
- Mouse gene regulatory network: https://networkrepository.com/bio-mouse-gene.php

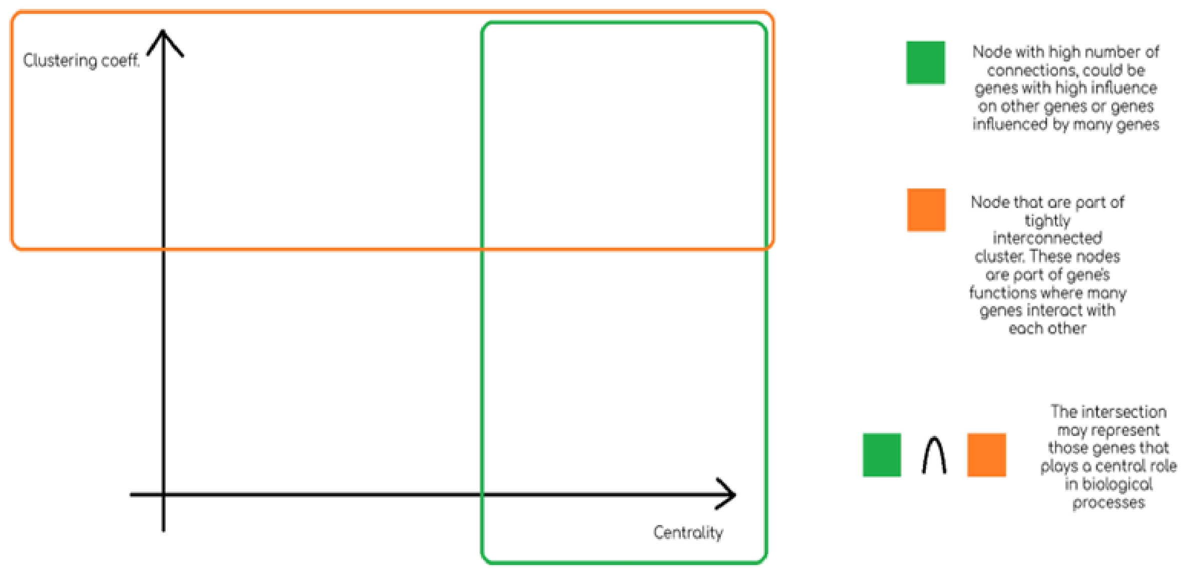
Graph analysis functions were used from the networkit and networkx libraries.
Additional libraries used are matplotlib for plotting graph results and numpy.
We included a setup.py file that automatically downloads the datasets and modifies them to be correctly read by the code.
Detailed description on how to run the code in the README.md file.
Repo hosted at: https://github.com/Coccoexe/LFN_Project

# Introduction

Mice serve a crucial function in the realm of biomedical research, therefore we are interested in finding similarities and differences between human and mouse genes regulatory networks.

In the regulatory network a node is a gene, and an edge is the interaction between two genes. We want to find a similarity between human and mice biological processes by looking at gene's importance and influence in the network.

In particular we set out to plot centrality and clustering coefficients for each node of both networks in a scatter chart: nodes that present high values for both metrics are to be set aside and further analyzed.


Clustering coeff.
Centrality
Node with high number of connections, could be genes with high influence on other genes or genes influenced by many genes
Node that are part of tightly interconnected cluster. These nodes are part of gene's functions where many genes interact with each other
∩
The intersection may represent those genes that plays a central role in biological processes

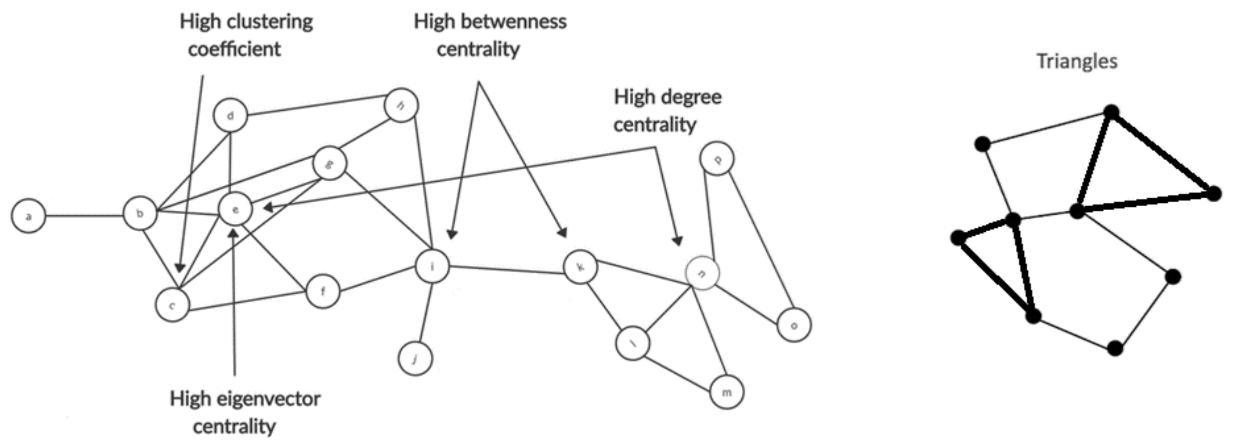
As you'll see in the 'Results' chapter things didn't apply in practice as we hoped they would in theory, and we ended up needing to discard the above mentioned view to analyze betweenness centrality, eigenvector centrality, triangle count and visualize them alone in histograms and together in a radar plot.

# Method / algorithms

To find the similarities, we compute several metrics:

- **Degree Centrality**: describes a node based on the number of connections it has. In our specific case a node with high centrality might be a gene with high influence on other genes.
- **Clustering Coefficient**: "is a measure of the degree to which nodes in a graph tend to cluster together."[1][1]
- **Betweenness Centrality**: "is a measure of centrality in a graph based on shortest paths. For every pair of vertices in a connected graph, there exists at least one shortest path between the vertices such that either the number of edges that the path passes through (for unweighted graphs) or the sum of the weights of the edges (for weighted graphs) is minimized. The betweenness centrality for each vertex is the number of these shortest paths that pass through the vertex."[2]
- **Eigenvector Centrality**: "is a measure of the influence of a node in a connected network. Relative scores are assigned to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. A high eigenvector score means that a node is connected to many nodes who themselves have high scores."[3]
- **Triangles**: the triangle count algorithm counts the number of triangles for each node in the graph. A triangle is a set of three nodes where each node has a relationship to the other two.

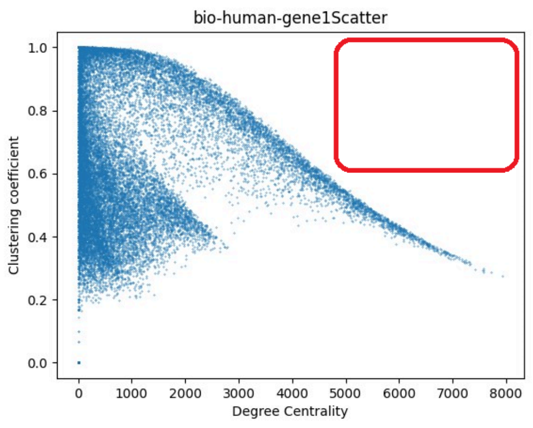
The algorithms used to compute these metrics are the functions defined in the networkit and networkx libraries. Degree centrality, clustering coefficient, betweenness centrality, eigenvector centrality were computed using networkit while the triangles were computed using networkx. Computed results were plotted into histograms, scatter charts and radar plots using matplotlib.

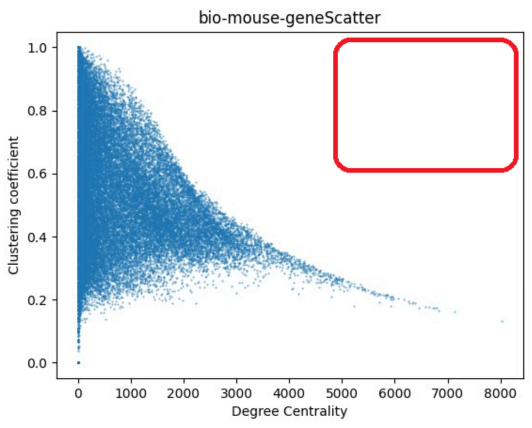

High clustering coefficient
High betweenness centrality
High degree centrality
High eigenvector centrality
Triangles
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p

[1][1]Clustering Coefficient, [2]Betweenness Centrality, [3]Eigenvector Centrality.

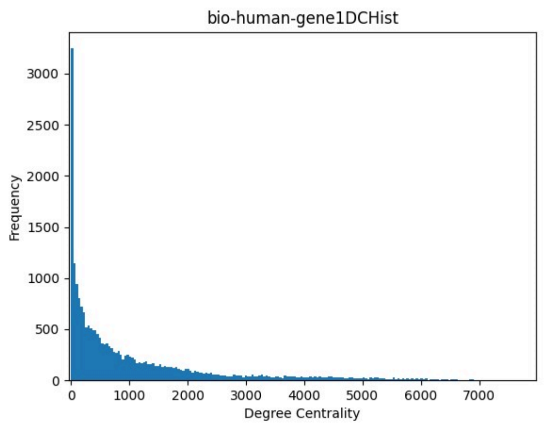
## Results

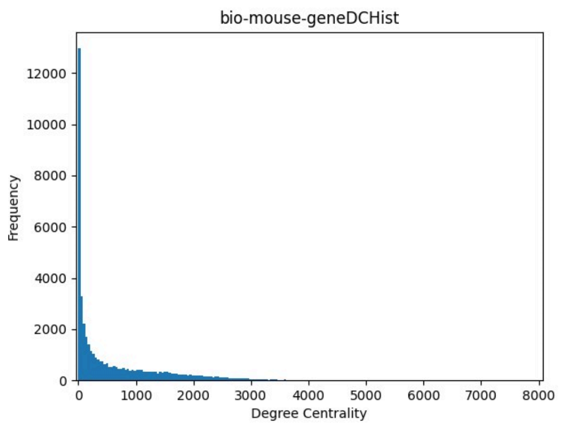
When actually plotting the centrality - clustering coefficients we obtained results that are not compatible with our initial goal of finding nodes with high values for both metrics.

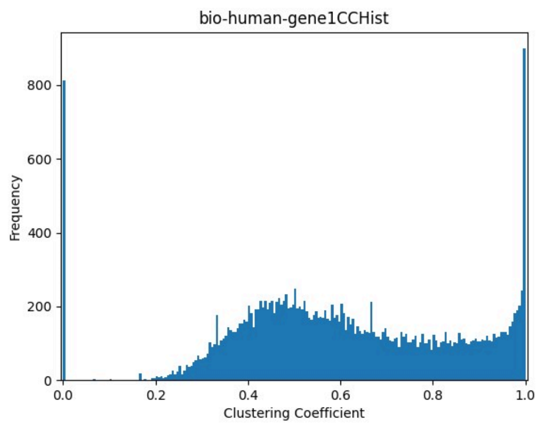


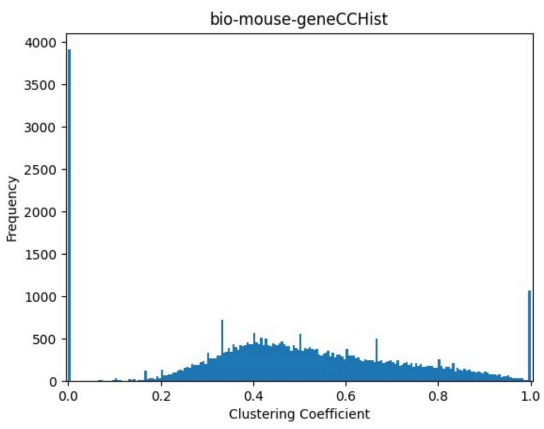


Separate histograms for both metrics are the following:

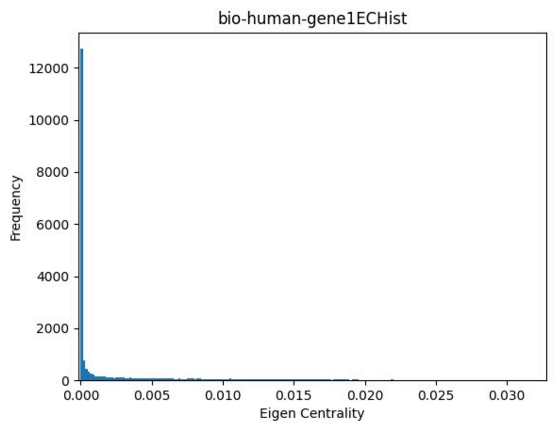


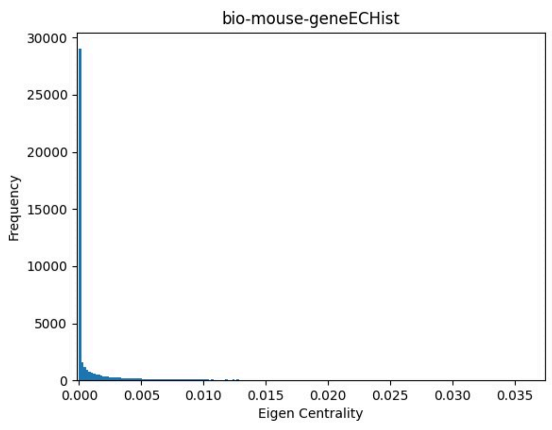


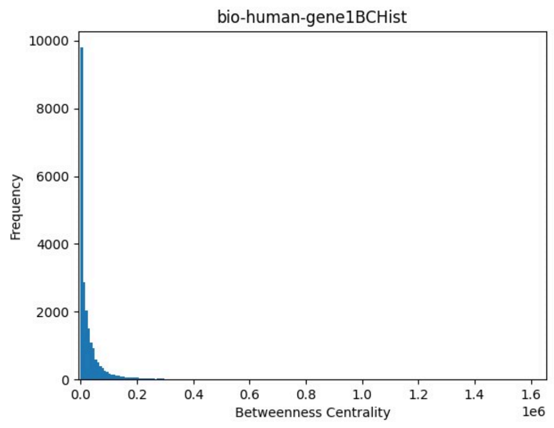


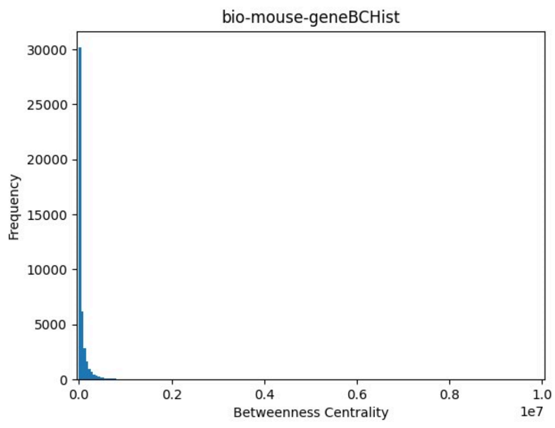


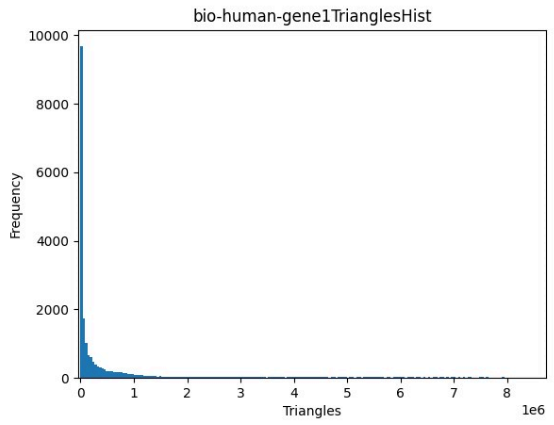
This lead us to computing other metrics we deemed possibly relevant for comparison, such as: eigenvector centrality, betweenness centrality and triangles.

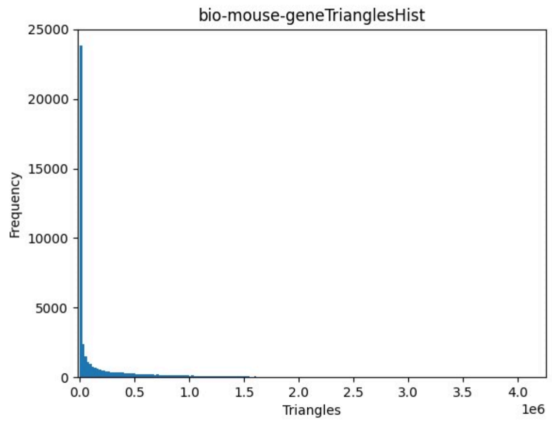






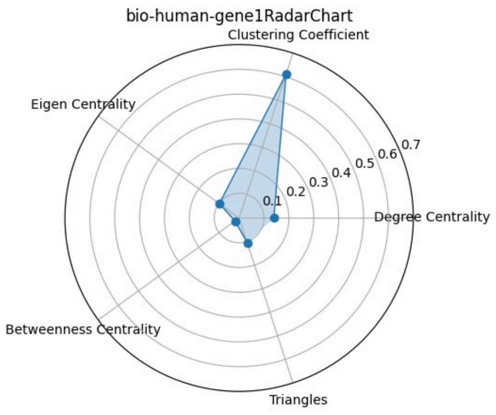


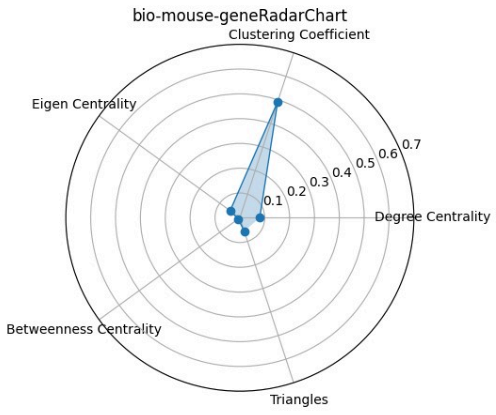




Again, we found very difficult to find and interpret meaningful differences between human and mouse gene regulatory networks, purely based on these statistics.

A final and more comprehensive overall view of our findings uses radar plots:





## Conclusion

In conclusion, many metrics didn’t reveal any significant structural difference between the two networks, which might indicate, as expected, a high degree of similarity between the two gene regulatory networks.
The only striking difference is located in the clustering coefficient histograms (and consequently in the centrality - clustering coefficients graph): there is a noticeable peak towards the higher range in the human network, that is not present in the mouse network.
This implies that there are more highly connected groups of genes and this could be due to biological, functional or structural reasons such as: the need for more highly-specialized gene clusters, the presence of more regulatory elements, or this could just be due to evolutionary heritage.
Further analysis by experts of the biology field might be able to provide a satisfactory explanation and determine if such difference is relevant and has real-world implications.

## Contributions

The initialization in the main file was made together and consists in reading the dataset with basic controls. We then planned the structure of the project, the main function is used only to call other functions and to print console information.

We worked together using the Live Share feature present in vscode for writing code. Overall we always brainstormed ideas and worked on their implementation together and we feel like every member contributed roughly equally to the project, but in case we need to provide more accurate details:

Alessio Cocco:
Code:

| | |
|---|---|
| • setup.py | - downloads and prepares the datasets we used. |
| • degreeCentrality() | - computes degree centrality |
| • clusteringCoefficient() | - compute cluster coefficients |
| • plotCombinedMetrics() | - plot a scatter chart with degree (x) and clustering (y) |

Ideas:
contributed to the idea of calculating degree and clustering to compare the two graphs by plotting the scatter chart to find interesting points.

Andrea Valentinuzzi:
Code:

| | |
|---|---|
| • betweennessCentrality() | - compute betweenness centrality |
| • eigenCentrality() | - compute eigenvector centrality |
| • plotHistogram() | - plot histogram given a metric |

Ideas:
contributed to the idea of computing other 2 metrics that focus not only on the importance of the node but also on the neighborhood. The idea was to plot all the different metrics to find out some shared characteristics between the two graphs.

Giovanni Brejc:
Code:

| | |
|---|---|
| • countTriangles() | - count triangles in the graph |
| • plotRadarChart() | - compute mean of metrics and plot a radar chart |
| • saveToFile() | - save a metric to file to avoid recomputation |
| • loadFromFile() | - load a metric from file |

Ideas:
contributed to the idea to calculate graphlets to search for a structural similarity. Since we couldn't find algorithms for graphlets count, only triangles were computed. Also thought about combining all metrics calculated in a radar chart to compare the overall view of the graphs properties.