Analysis of Methods for Automatic Graph Clustering

Milestone 1 Report

Randa Elshafei

Avery Peiffer

I. Introduction

Computational modeling is a useful technique for capturing the dynamics of complex systems; specifically, many systems can be modeled as graphs of nodes and edges. However, creating such a graph is not trivial. Accurately modeling a biological system, for example, requires a steep up-front investment in collecting useful information. This process can involve consulting domain experts for background information and conducting experiments in a wet lab, in addition to searching through hundreds of papers, therefore drastically increasing the complexity of creating and extending models. It is not feasible to rely on manually curated models in every case, as researchers do not possess the time or resources to do so. For this reason, several methodologies aim to automate this graph creation process by using tools to extract relevant events from literature. Of course, not all data extracted from literature can be used as a direct input for these tools; instead, one or more clustering methods must be applied first to extract relevant data and increase the tool's efficiency.

II. Background

For this project, we have chosen to work with the graph automation tool CLARINET, which extracts information from literature and creates a directed graph to model intracellular signaling [1]. CLARINET has as inputs a baseline model and a machine reading output of literature. The tool then creates an event collaboration graph (ECLG), where nodes represent the distinct events extracted from the literature and edges represent the co-occurrence of the two nodes. CLARINET uses two types of assessment to label nodes: individual assessment (node assessment), and pair assessment (edge weights). The individual and pair assessments are calculated using a frequency class metric, like that used in computational linguistics. From there, the ECLG is clustered based on edge weights, finding the most relevant clusters for answering the initial literature search. Overall, the model can quickly assemble and extend a model, collecting existing information with little time overhead.

III. Project Progress

Our focus for this introductory milestone was on downloading and running the existing CLARINET materials. The CLARINET repository includes both a Jupyter Notebook and a Python script that can be executed to run the code. We first attempted to both run the code from the Jupyter Notebook for ease of use and clarity in explaining the code.

A. Problems we faced

Randa was able to generate the output files that were originally generated by Yasmine but was not able to plot the same graph on her laptop (explained in the following paragraph) using the Jupyter Notebook.; however, for reasons still unknown, Avery had errors when trying to execute the entire file. The issue seems to be related to the installation of the community package, but we were not able to fix it despite speaking to Yasmine about the error. Instead, Avery was able to run the Python script to get the same output. There were some minor issues relating to running the code, but we were able to maneuver the script to a workable state thanks to Yasmine's help. The output from running the CLARINET code is shown below in Figure 1. These clusters represent the co-occurrence of events, either excitatory or inhibitory, in the same paper.

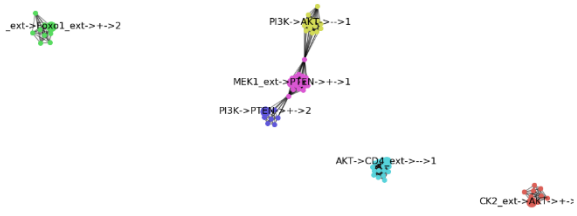

Figure 1: The clustering output from running the CLARINET code after removal of less frequent events

Originally, Randa encountered an issue with plotting the output, wherein the output files generated by her Jupyter Notebook were different from those generated by Yasmine. There was no obvious solution on Jupyter Notebook, so Yasmine eventually suggested commenting out the plotting function as it does not affect the actual results. She instead suggested using a visualization tool to display the network before and after cluster merging using CLARINET. The Cytoscope tool was used to visualize the network. Cytoscope takes a text file of the network representation and generates the network graph when source and target nodes are specified. Figures 2 and 3 are the Cytoscope visualizations for the case presented in the CLARINET paper.

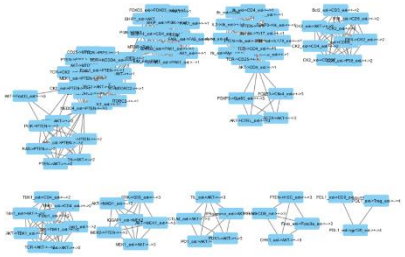

Figure 2: The clustering output from running the CLARINET code before removal of less frequent events

Figure 3: The clustering output from running the CLARINET code after removal of less frequent events

B. Results using CLARINET for new cases

We asked Yasmine to provide us with some new cases for running CLARINET. We have successfully generated results for two new cases. Yasmine did not provide results for these cases. Considering that we got the same results for the original case from the paper, we are assuming that these results are also correct.

C. Future plans

For future milestones, we aim to continue studying and analyzing CLARINET's outputs given the inputs used in the original paper as well as the two new cases provided by Yasmine. We will remain in communication with Yasmine to understand the inner workings of CLARINET, and how we can extend this knowledge to other methodologies in graph clustering. As stated in our proposal, we aim to read and analyze papers that take different approaches to automatic graph clustering. We still plan to explore the Git tool, which uses intensity topology to cluster a graph [2]. We will also work with the yWorks tool, which guides us on clustering graphs using different methods [3]. These tools will help us to gain a greater understanding of graph clustering on a greater scale, instead of within the scope of the CLARINET tool.

The following figure illustrates the paths we are going to explore for this project.

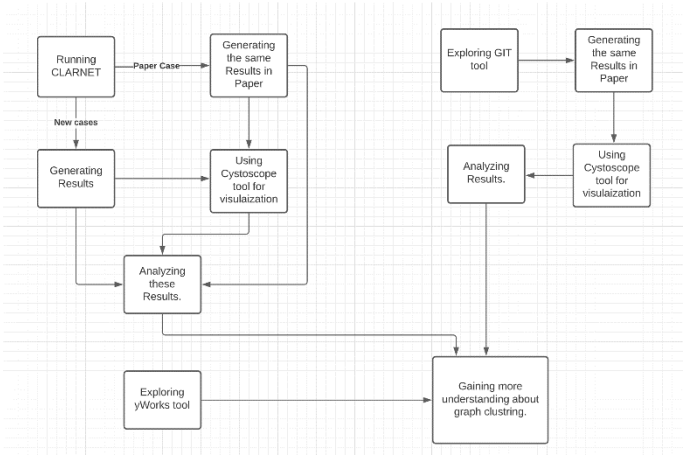

Figure 4: Flowchart diagram representing our project plan

REFERENCES

[1] Y. Ahmed, C. A. Telmer, and N. Miskov-Zivanov, "Clarinet: Efficient learning of dynamic network models from literature," *OUP Academic*, 03-Jun- 2021. [Online]. Available: https://doi.org/10.1093/bioadv/vbab006. [Accessed: 09-Feb-2022].

[2] Z. Gao, H. Lin, C. Tan, L. Wu, and S. Z. Li, "Git: Clustering based on graph of intensity topology," *arXiv.org*, 04-Oct-2021. [Online]. Available: https://arxiv.org/abs/2110.01274. [Accessed: 09-Feb-2022].

[3] "YFiles for HTML demo applications - clustering algorithms," *yWorks, the diagramming experts*. [Online]. Available: https://live.yworks.com/demos/analysis/clustering/. [Accessed: 09-Feb-2022].