Analysis of Methods for Automatic Graph Clustering

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 an easy feat. 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

One such graph automation tool is 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 Scope

In this project, we wish to explore the diverse ways in which graphs are automatically clustered. We will explore the CLARINET tool more, understanding and analyzing the outputs given the inputs used for the paper. By analyzing new case studies and tools, we can explore possible improvements for the performance of the CLARINET tool. For example, a new filtering criterion that selects fewer events initially could be applied to the input, improving the overall performance.

We also wish to explore other methodologies used in graph clustering by reading papers that take different approaches to automatic graph clustering. One such tool that we would like to explore is called Git, which clusters based on a graph of intensity topology [2]. We will start by working with a tool called yWorks, which will show us how to cluster graphs using different methods [3]. This tool will help us understand the novel methodologies proposed in the literature, such as CLARINET and Git.

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].