

App Development for Analyzing Biological Networks

Students: Leonardo Villa, Valeria Garcia, William Lopez, Tyler Uwate, Oscar Alvarez
Mentor: Ananda Mondal, Florida International University
Instructor/Faculty: Masoud Sadjadi, Florida International University

INTRODUCTION

Cytoscape, a bioinformatics application, visually examines biological networks, including those related to cancer. Its CyFinder extension integrates clustering algorithms to identify diseases in their early stages. Another feature called SubgraphFinder is responsible for creating and applying these algorithms, thus improving Cytoscape's capabilities for research purposes.

SYSTEM DESIGN

This app uses Cytoscape and CyFinder and it's developed primarily in Java and is designed with components dedicated to processing and analyzing biological network data.

```
private void Coarse_Tuning(Graph g){  
    // for (module Mi: Map M)  
    Map<Node, Integer> previousNodeToModule = new HashMap<>();
```

VERIFICATION

- The testing of the algorithm aims to verify the following:
- Accurate implementation of the algorithm across different graph.
 - Correct operation across a range of network file formats and data structures.

CURRENT SYSTEM

The system supports converting directed graphs to undirected ones through additive or transformative methods, allowing users to add edges between nodes or transform edge objects for compatibility with Cyfinder. It includes customizable search conditions (bi-partite, clique, or directed clique), search algorithms (Breadth or Depth First Search), graph ordering criteria, and subgraph saving options.

DESIGN

Algorithm 5 Fine-tuning
1 function FINETUNE(Network G, Node-to-module map M)
2 M ← one module per node in G
3 Update M to mirror M
4 M ← REPEATEDNODEAGGREGATION(G, $\tilde{M}$) (Algorithm 3).
5 return M
Algorithm 6 Coarse-tuning
1 function COARSETUNE(Network G, Optional node-to-module map M)
2 for module m_i in M do
3 G_i ← network of nodes and links within m_i
4 Sub-map M_i ← REPEATEDNODEAGGREGATION(G_i) (Algorithm 3)
5 $\tilde{G}$ ← network of submodules M_i
6 $\tilde{M}$ ← map of nodes in $\tilde{G}$ to M
7 M ← REPEATEDNODEAGGREGATION($\tilde{G}$, $\tilde{M}$) (Algorithm 3).
8 M ← M($\tilde{G}$) such that M($\tilde{G}$) $\Leftrightarrow$ M($\tilde{G}$)
9 return M

SUMMARY

The goal of the project is to improve the current CyFinder application used for analyzing biological networks. It emphasizes revealing clique-like and bipartite-like patterns to gain deeper insights into disease progression at the protein network level. We have successfully implemented Algorithm 5 and 6 as part of this effort.

REQUIREMENTS

- Java 11.0
- Apache Ant
- Maven
- CytoScape 3.9.1
- Eclipse (Optional)



IMPLEMENTATION

Introduced course-tuning function, Infomap tries to move submodules out from existing modules into new or neighboring modules (Algorithm 6). First, Infomap treats each module as a network on its own and applies repeated node aggregation (Algorithm 3) on this network. This procedure generates one or more submodules for each module. Infomap then replaces the modules by the submodules and moves the submodules into their modules as an initial solution

REFERENCES

- Cytoscape.js API Documentation(2021). Retrieved from: https://cytoscape.org/javadoc/current_release/org/cytoscape/view/vizmap/VisualStyle.html
- Mapping Higher-Order Network Flows in Memory and Multilayer Networks with Infomap (Daniel Edler, Ludvig Bohlin and Martin Rosvall).

ACKNOWLEDGEMENT

The material presented in this poster is based upon the work supported by Dr. Ananda Mondal. We thank Dr. Ananda Mondal for his assistance and mentorship that we received throughout the senior design project.