

CyFinder: Analyzing Biological Networks

Students: Kyle Pulido, Rolans Apinis

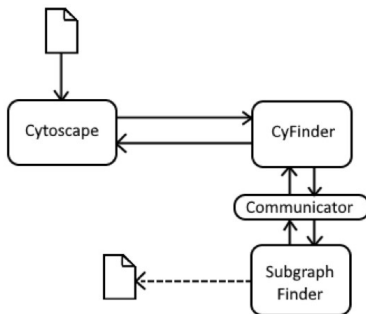
Mentor: Raihanul Bari Tanvir, Dr. Ananda Mohan Mondal

Instructor/Faculty: Dr. Masoud Sadjadi, Florida International University

Problem

Cytoscape is a program used for working with data and displaying networks. Sometimes, users need to identify bipartite and clique subgraphs within their network in order to draw meaningful conclusions from their data. However, Cytoscape does not include this functionality by default.

CyFinder is a plugin that does just that. However, it is still missing some features, and has a handful of issues. Within this project we try to improve CyFinder further by adding functionality and fixing up other issues that were left unfinished by the previous team.



Current System

Networks currently need to be in an undirected state to be processed. After converting a graph to an undirect state with one of two methods (additive or transformative), the user can select various options to locate subgraphs.

Currently, the system supports displaying the graphs in the window or saving subgraphs to a file. It also supports sorting graph outputs and doing a BFS traversal in order to display the resulting graph as a tree.

System Design

Cytoscape reads graphs in from a file. It then makes these graphs accessible to the "CyFinder" module, which is a GUI module intended to tell the actual SubgraphFinder what to do. It does this through a singleton Communicator class that links the two.

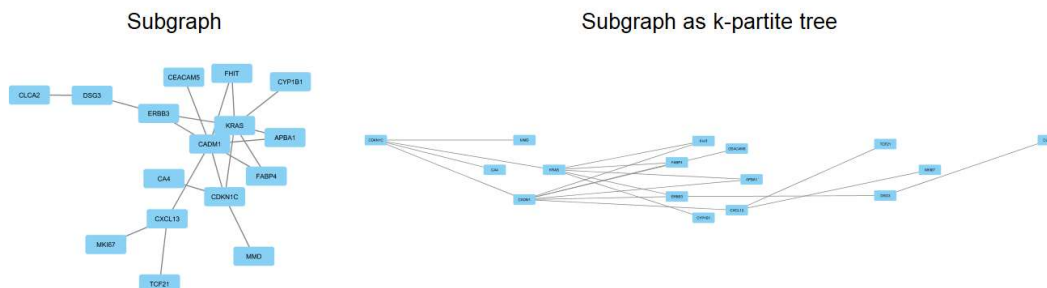
SubgraphFinder is the class that contains the actual logic for finding subgraphs. It uses its own simplified Graph class for its tasks, and then converts it back into a Cytoscape graph when done, passing it back through the Communicator and to CyFinder/Cytoscape. Alternatively, SubgraphFinder can write directly to an external file.

Object Design

CyFinder uses several different objects for its functions – too many to list and include. Some of the most important root classes are listed here.

- **ConfigDialog**: Swing GUI dialog that allows the user to choose properties of the algorithm for the SubgraphFinder to user.
- **Communicator**: Singleton class that exists to pass lists of data (including Graphs) to CyFinder.
- **CytogrouperMain**: Handles assigning group values to nodes in order to allow them to display like in the screenshot below.
- **Condition**: Condition classes check if graphs satisfy a condition.
- **Algorithm**: Class instantiated by ConfigDialog to move information and arguments. Uses Bundles for this purpose.
- **Bundle**: Passes information and arguments about the algorithm to use, as opposed to Communicator, which passes graphs.
- **Graph/Edge/Node**: Used to represent a graph in memory. An isolated implementation that is completely independent of the Cytoscape API.
- **CytogrouperEdge/CytogrouperNode**: Used to represent a Cyfinder graph. Simpler than the normal Edge/Node classes, and the CytogrouperNode class in particular contains information that will be displayed within Cytoscape, such as partition numbers.
- **ConditionedGraph**: Graph subclass that continually checks for a Condition.
- **GraphReader/GraphWriter**: Interfaces used for reading and writing graphs from/to files. Has subclasses **SIFGraphIO** and **SimpleGraphIO**.

Screenshot



Requirements

- This app requires Cytoscape 3.7.2. 3.8 may work, however it was released during development and has not been tested.
- The CyFinder app should be installed through the Cytoscape App Manager.
- The data inputted must be in the .csv or .txt file format.
- Data must then be pre-processed using the options available in the "Make Undirected" tab in CyFinder.
- Subgraph Finder needs to be opened. Then the search condition and a search algorithm as well as output format must be selected to display subgraphs.

Implementation



- Java: Language used to create CyFinder
- Maven: Combines the logic with the GUI to generate the CyFinder JARs
- Apache Ant: Responsible for building the project that contains the logic of CyFinder
- Cytoscape API: Necessary tools needed to implement Cytoscape features for CyFinder

Verification

Manual testing was used to test each new feature added, as the plugin nature of Cyfinder makes isolated testing as-of-yet impossible.

Summary

Currently, CyFinder is capable of locating clique and bipartite graphs. Newly added features include the ability to assign partition numbers to make k-partite graphs, to display the graphs as trees, and to be able to properly save the graphs to a file, among other things. Whenever possible, this code was documented heavily.

As it stands, CyFinder still has some issues, such as causing errors when certain options are chosen, and a few other wanted features which we leave for future teams.

Acknowledgement

This work was supported by Dr. Ananda Mondal. The material presented in this poster is based upon the work created by Cornelia Ada Schultz at Claflin University. We also would like to thank Raihanul Tanvir for his assistance.