

CyFinder: Analyzing Biological Networks

Students: Rolans Apinis, Kyle Pulido

Mentor: Raihanul Bari Tanvir, Dr. Ananda Mohan Mondal

Instructor: Dr. Masoud Sadjadi, Florida International University

Problem

Cytoscape is a software that deals with biological networks and has many plugins that add all kinds of functionality to the software. Our current project CyFinder allows users to perform different actions on undirected graphs, however the given implementation still has several features that are not completely integrated, which can cause incorrect outputs or, worst case, freeze up your entire device.

The goal of this project is to both to fix the currently not working functions as well as add new ones, at the same time upgrading the UI to help improve flow.

Current System

The current system allows the user to convert directed graphs to undirected graphs using two approaches: additive, which involves the adding another edge between any two connected nodes, or transformative, which changes the given edge object into an object cyfinder can work with.

It also allows user to specify the search conditions (bipartite, clique or directed clique), select the search algorithm they want to use (Breath or Depth First Search), if they want the graphs to be ordered (based on edge weight and SUID) and how they want to save the generated subgraphs (add truth table, generate separate subgraphs)

Requirements

Hardware:

- No specific hardware was used during the development of the application.

Software:

- Java 8.0 Maven
- Apache Ant
- Cytoscape API
- Eclipse Oxygen 3.0A
- Cytoscape 3.7.2

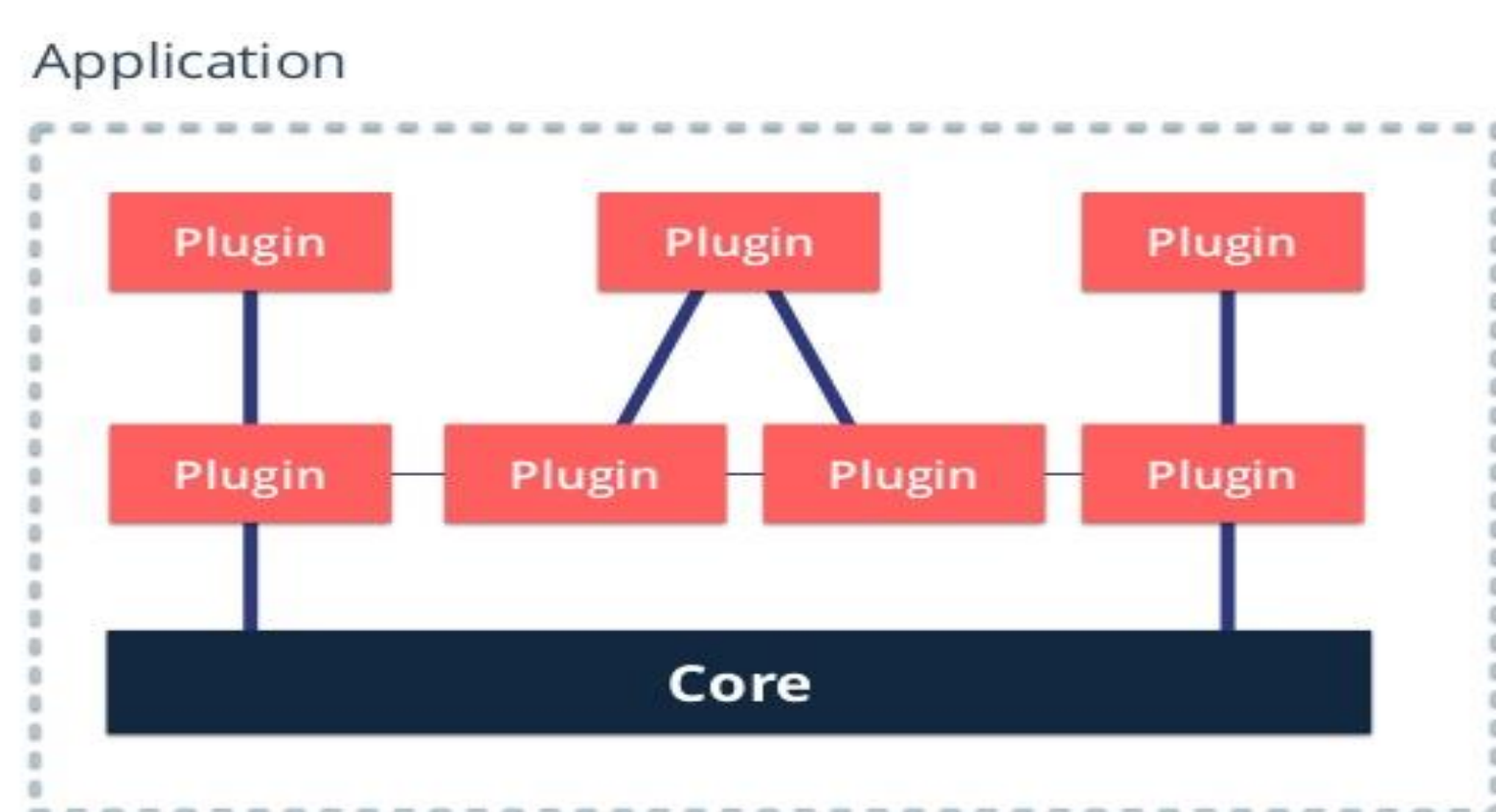
Version Control

- GitHub (<http://www.github.com>).



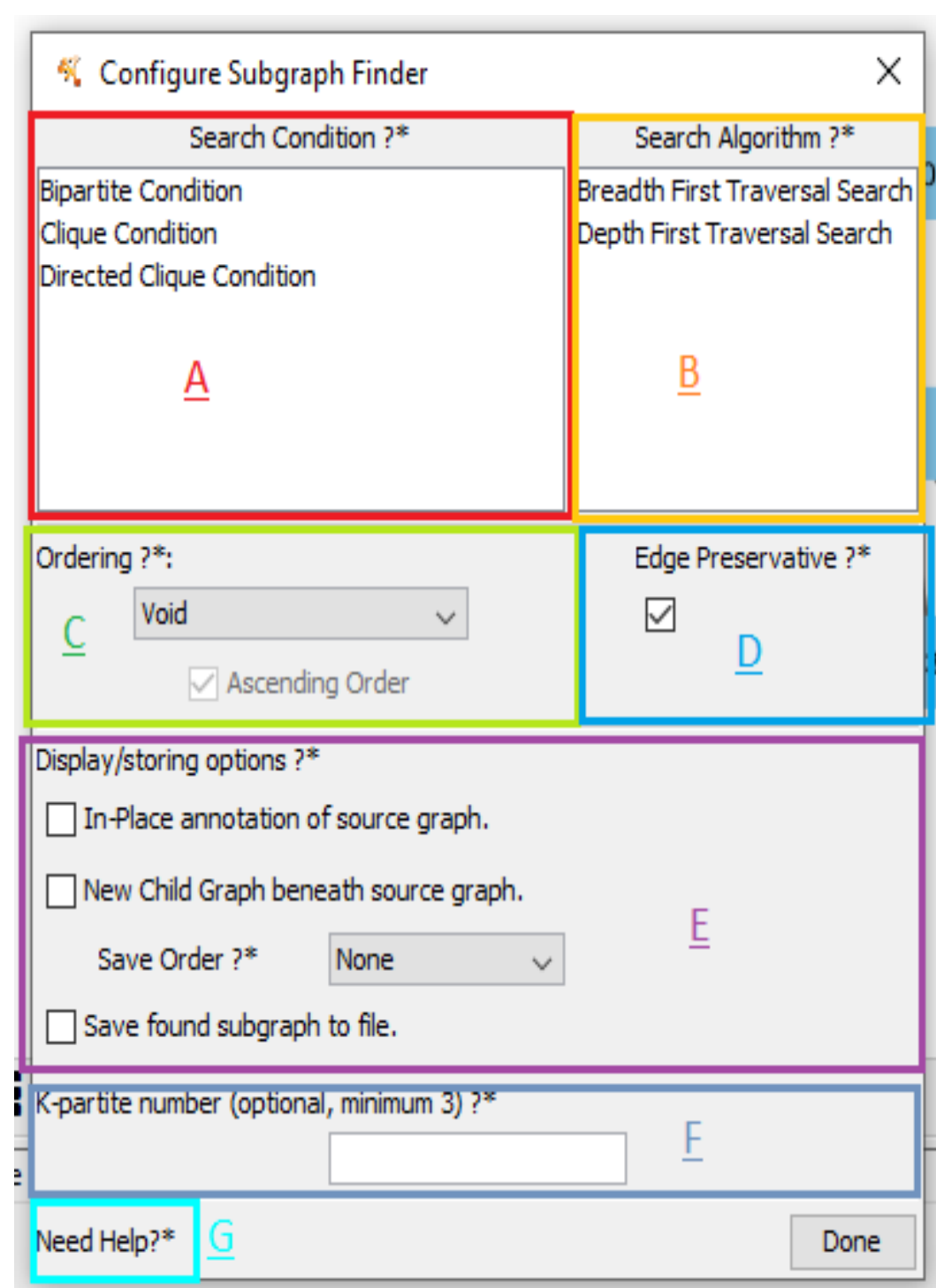
System Design

Plugin-based software architecture

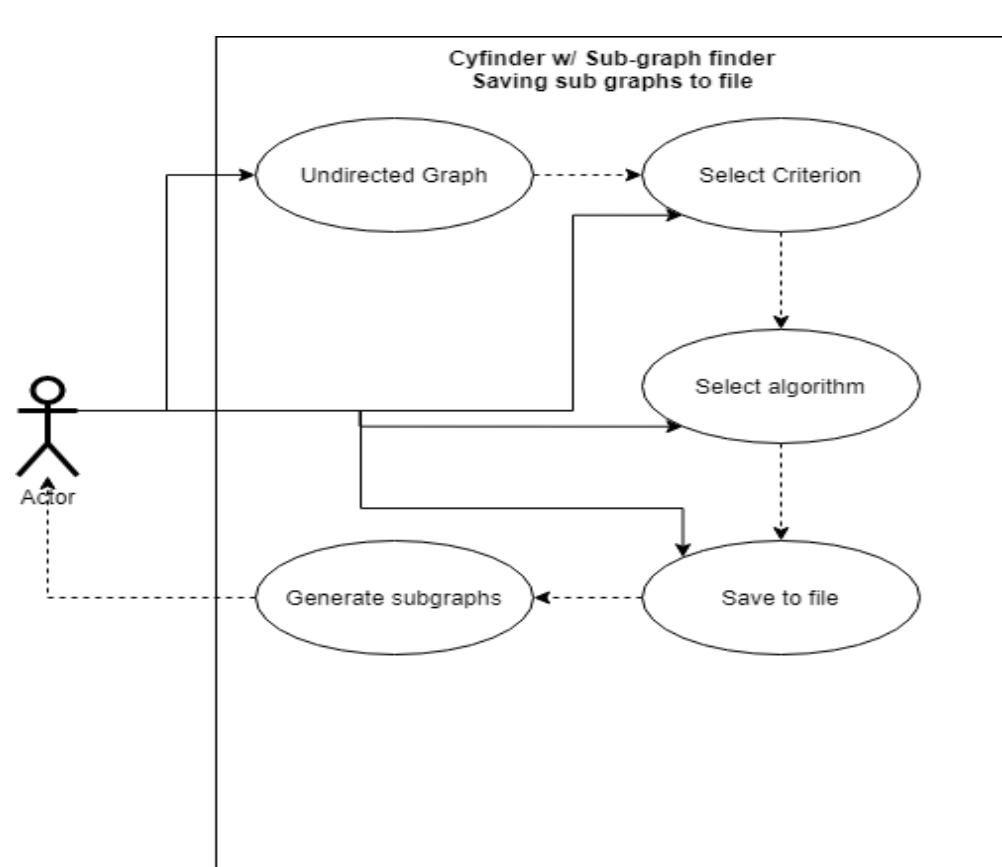
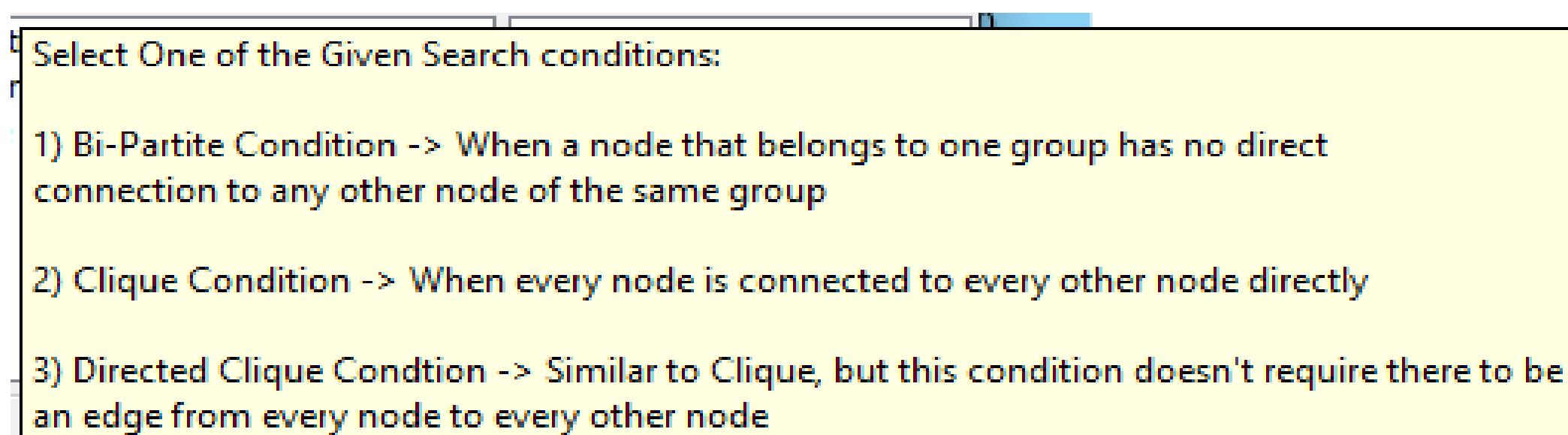


New UI

Screenshots



- A) Search Condition
- Bipartite
 - Clique
 - Directed Clique
- B) Search Algorithm
- Breadth First Traversal Search
 - Depth First Traversal Search
- C) Ordering based on edges
- Edge Weight Comparator
- D) Edge Preservation
- E) Select how to store the generated results
- In-Place Annotation
 - New Child Graph
 - Can select ordering based on nodes
 - Save found subgraphs to file
- F) K-partite number
- G) Need Help



Verification

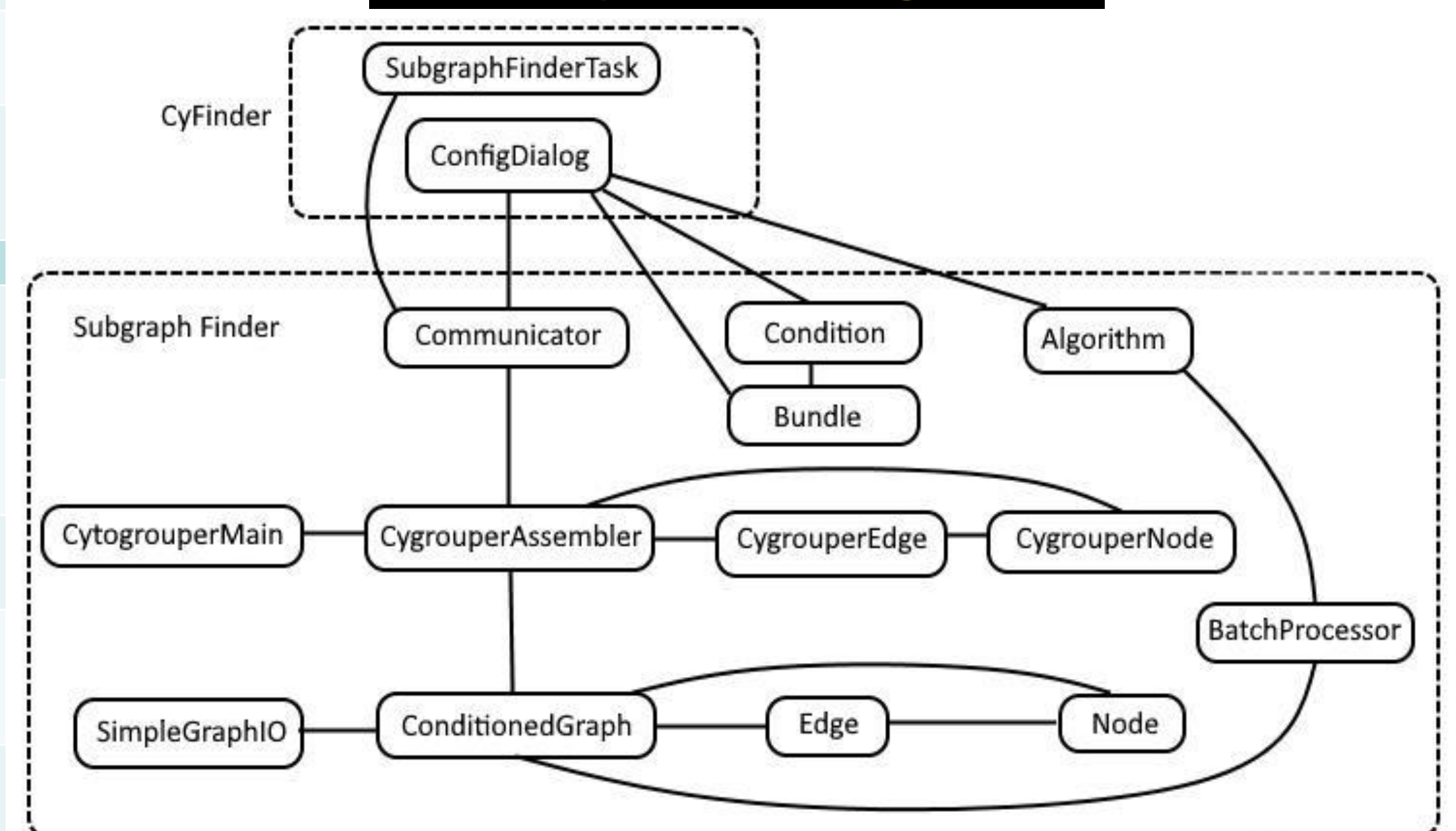
I	Expected Results	Observed Results
Transformative Undirected Graph		
Bipartite + BFS + edge preservation on	Prints bipartite subgraphs	Prints bipartite subgraphs
Bipartite + BFS + edge preservation off	Prints bipartite subgraphs that contain all original nodes	Prints bipartite subgraphs that contain all original nodes
Bipartite + DFS + edge preservation on	Prints bipartite subgraphs	Prints bipartite subgraphs
Bipartite + DFS + edge preservation off	Prints bipartite subgraphs that contain all original nodes	Prints bipartite subgraphs that contain all original nodes
Clique + BFS + edge preservation on	Prints clique subgraphs	Prints clique subgraphs
Clique + BFS + edge preservation off	Prints clique subgraphs that contain all original nodes	Prints subgraphs consisting of a pair of nodes
Clique + DFS + edge preservation on	Prints clique subgraphs	Prints clique subgraphs
Clique + DFS + edge preservation off	Prints clique subgraphs that contain all original nodes	Prints subgraphs consisting of a pair of nodes
Directed Clique + BFS + edge preservation on	Prints directed clique subgraphs	Prints directed clique subgraphs
Directed Clique + BFS + edge preservation off	Prints directed clique subgraphs that contain all original nodes	Prints subgraphs consisting of a pair of nodes
Directed Clique + DFS + edge preservation on	Prints directed clique subgraphs	Prints directed clique subgraphs
Directed Clique + DFS + edge preservation off	Prints directed clique subgraphs that contain all original nodes	Prints subgraphs consisting of a pair of nodes
Additive Undirected Graph		
Bipartite + BFS + edge preservation on	Prints bipartite subgraphs	Prints bipartite subgraphs
Bipartite + BFS + edge preservation off	Prints bipartite subgraphs that contain all original nodes	Prints bipartite subgraph with all original node
Bipartite + DFS + edge preservation on	Prints bipartite subgraphs	Prints bipartite subgraphs
Bipartite + DFS + edge preservation off	Prints bipartite subgraphs that contain all original nodes	Prints bipartite subgraph with all original node
Clique + BFS + edge preservation on	Prints clique subgraphs	Prints subgraphs consisting of a pair of nodes
Clique + BFS + edge preservation off	Prints clique subgraphs that contain all original nodes	Produces empty files as output for each node in the given graph, does not throw error (included warning)
Clique + DFS + edge preservation on	Prints clique subgraphs	Prints subgraphs consisting of a pair of nodes
Clique + DFS + edge preservation off	Prints clique subgraphs that contain all original nodes	Produces empty files as output for each node in the given graph, does not throw error (included warning)
Directed Clique + BFS + edge preservation on	Prints directed clique subgraphs	Prints directed clique subgraphs
Directed Clique + BFS + edge preservation off	Prints directed clique subgraphs that contain all original nodes	Prints subgraphs consisting of a pair of nodes
Directed Clique + DFS + edge preservation on	Prints directed clique subgraphs	Prints directed clique subgraphs
Directed Clique + DFS + edge preservation off	Prints directed clique subgraphs that contain all original nodes	Prints subgraphs consisting of a pair of nodes

Implementation

- Java 8.0** - Main language used to create CyFinder.
- Maven** - Combined the logic with the GUI to generate JAR.
- Apache Ant** - Responsible for building the project that contains the logic of CyFinder.
- Cytoscape API** - Necessary tools needed to implement Cytoscape features for CyFinder.
- Cytoscape 3.7.2** – core software



Object Design



Summary

- Fixed issues with infinite loop and erroring when edge preservation was disabled
- Added warnings when users try to use try to use an unsupported/nonfunctioning criterion
- Updated UI for SubGraph Finder
- Added descriptions to each option to make software easier to use for regular users
- Did extensive testing to show functionality of the whole system

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