

Florida International University
School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title: Analyzing Biological Networks

Team Members: Ariel Sari, Evyatar Saia

Product Owner(s): Ananda Mohan Mondal

Mentor(s): Raihanul Bari Tanvir, Ananda Mohan Mondal

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

A group of proteins or genes work together by forming a cluster, which could be a clique or clique-like structures, to accomplish different functions. These functions could be beneficial or adversarial such as initiation or progression of a disease. Usually, for a cancer to occur, a group of proteins work together (forming a clique-like structures) to initiate the disease. Then the transfer of disease signal to the next group of proteins happens by forming a bipartite graph. This process continues from initiation to stage to stage.

This document presents the information necessary to gain a good understanding of CyFinder, which is an app developed for the software Cytoscape. The main objective of CyFinder is to identify the clique-like and bipartite-like structures from a biological network such as protein-protein interaction network, which would represent disease progression at protein network level.

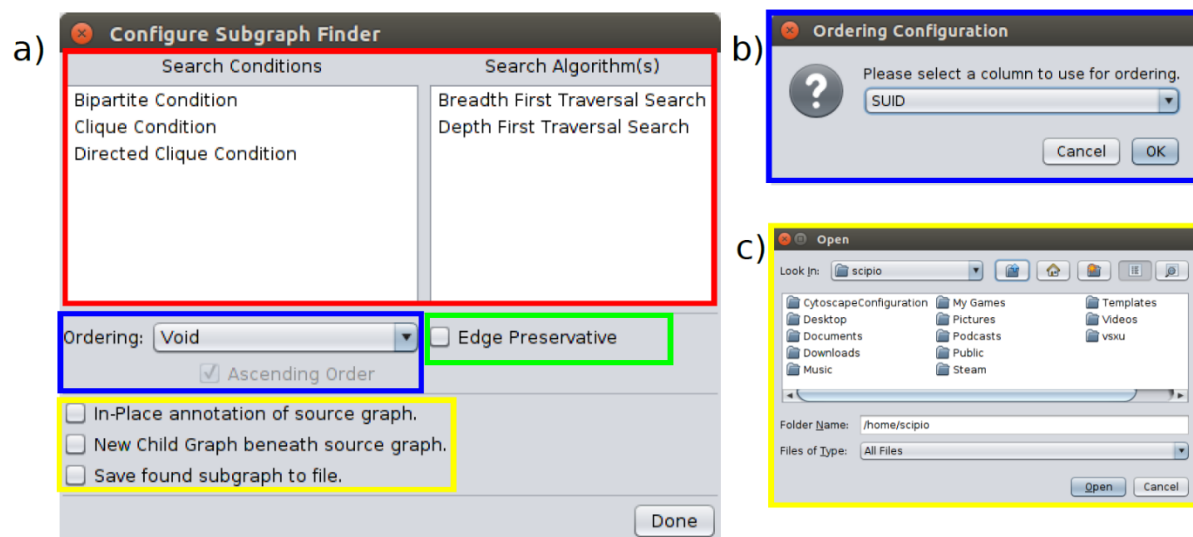
Table of Contents

INTRODUCTION	5
CURRENT SYSTEM	5
PURPOSE OF NEW SYSTEM	6
USER STORIES	
IMPLEMENTED USER STORIES	6
PENDING USER STORIES	8
PROJECT PLAN	
HARDWARE AND SOFTWARE RESOURCES	9
SPRINTS PLAN	10
<i>Sprint 1</i>	10
<i>Sprint 2</i>	11
<i>Sprint 3</i>	12
<i>Sprint 4</i>	13
<i>Sprint 5</i>	15
<i>Sprint 6</i>	16
SYSTEM DESIGN	
SYSTEM AND SUBSYSTEM DECOMPOSITION	19
APPENDIX	21
APPENDIX A - UML DIAGRAMS	21
<i>Static UML Diagrams</i>	21
<i>Dynamic UML Diagrams</i>	21
APPENDIX B - SEQUENCE DIAGRAMS	23
APPENDIX C - USER INTERFACE DESIGN	25
APPENDIX D - USER MANUALS, INSTALLATION/MAINTENANCE DOCUMENT, SHORTCOMINGS/WISHLIST DOCUMENT AND OTHER DOCUMENTS	26
REFERENCES	36

INTRODUCTION

Cytoscape is a popular software 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. The problem is that Cytoscape does not have a feature that can do that by default. Therefore, there is a need for a plugin that can do that while also making the subgraphs easy to visualize.

Current System



Currently, networks need to be in an undirected state to be processed. The current system provides 2 ways of doing this, the additive method and the transformative method. The type of subgraph to be search for is then specified, bipartite or clique. One of two algorithm is then provided for the user to choose from, breadth first search or depth first search. Then, a way of saving the result is specified.

Purpose of New System

Cytoscape is often used by researchers for analyzing large biological networks where each node within the network represents a gene. It is beneficial to parse and organize these networks in different manners in order to obtain different meaning from the data. Two particularly meaningful organizations of the data are cliques and bipartite arrangements. In a bipartite graph there exists two disjoint and independent sets, while in a clique every node is adjacent to the other. It's particularly useful to study these conditioned graphs when studying disease progression. In the new system, there exists a new way of visualizing and working with the bipartite and clique subgraphs which makes it easier for researchers to work with these networks.

USER STORIES

The following section provides the detailed user stories that were implemented in this iteration of the Analyzing Biological Network project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

User Story 1: As a developer, I want to familiarize myself with the additive method feature

User Story 2: As a developer, I want to familiarize myself with the transformative method feature

User Story 3 & 4: As a developer, I want to learn about the frameworks and technologies used in CyFinder

User Story 5: As a developer, I want to analyze the transformative method for undirectedness

User Story 6: As a developer, I want to analyze the BFS algorithms from the source code

User Story 7: As a developer, I want to analyze the DFS algorithm from the source code

User Story 8: As a developer, I want to study the edge comparator class to determine if it can filter networks by weight

User Story 9: As a developer, I want to test the additive vs transformative method for pre-processing code

User Story 10: As a developer, I want to fix breadth first search algorithm code

User Story 11: As a developer, I want to attach node attribute data to the networks in Cytoscape

User Story 12: As a developer, I want to fix the .pom file for successful Maven integration with CyFinder

User Story 13: As a developer, I want to create an interface for communication between the CyFinder module and subgraph finder module

User Story 14: As a user, I want to know how to make my network undirected

User Story 15: As a developer, I want to create a singleton class that would manage shared data between modules

User Story 16: As a developer, I want to implement a function that reads the names of the genes from imported data

User Story 17: As a developer, I want to implement a function that parses the imported data into the correct format

User Story 18: As a developer, I want to implement a function that writes the data into the node table in Cytoscape

User Story 19: As a user, I want to know how to use the bipartite condition on my network

User Story 20: As a developer, I want to modify the node tables internal attributes to make it subgraph dependent

User Story 21: As a developer, I want to generate a .jar file from the subgraph finder module which contains .class files along with its corresponding source code

User Story 22: As a developer, I want Maven to include local dependencies in its build path

User Story 23: As a developer, I want to generate a .jar file from the CyFinder module using Ant-builder

User Story 24: As a developer, I want to a way of converting node objects into CyGrouperNode objects

User Story 25: As a user, I want to know how to use the clique condition on my network

User Story 26: As a developer, I want to integrate the CyGrouper package into the subgraph finder module

User Story 27: As a developer, I want to convert the source-target relationship of genes into a Java Map object

User Story 28: As a developer, I want to convert a Java Map object containing the names of the genes into a tree of CyGrouperNodes

User Story 29: As a developer, I want to apply the breadth first search algorithm to group nodes into two distinct sets

User Story 30: As a developer, I want to create a group column in the node table for each subgraph which contains each set a node belongs to

User Story 31: As a developer, I want to map a Cytoscape GUI node into its tree entry

User Story 32: As a user, I want to view which group a node belongs to in a bipartite subgraph

User Story 33: As a user, I want to view which group a node belongs to in a clique subgraph

User Story 34: As a developer, I want to create documentation for other developers to use

Pending User Stories

User Story 35: As a user, I want to be able to view the largest complete graph within a bipartite subgraph

User Story 36: As a developer, I want to organize subgraphs in descending order

User Story 36: As a developer, I want to organize subgraphs in by its average weight

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

Java

- 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

Sprints Plan

Sprint 1

September 9th – September 20th

Total Stories

- **User Story 1:** As a developer, I want to familiarize myself with the additive method feature
- **User Story 2:** As a developer, I want to familiarize myself with the transformative method feature
- **User Story 3 & 4:** As a developer, I want to learn about the frameworks and technologies used in CyFinder

Ariel Sari

- **User Story 1:** As a developer, I want to familiarize myself with the additive method feature
- **User Story 3:** As a developer, I want to learn about the frameworks and technologies used in CyFinder

Evyatar Saias

- **User Story 2:** As a developer, I want to familiarize myself with the transformative method feature
- **User Story 4:** As a developer, I want to learn about the frameworks and technologies used in CyFinder

Sprint Review

Attendees: Ananda Mondal, Raihanul Tanvir, Ariel Sari, Evyatar Saias

Date: September 19th, 2019

Start Time: 5:00pm

End Time: 6:45pm

The following stories were accepted after the approval of the PO during the sprint review:

- **User Story 1:** As a developer, I want to familiarize myself with the additive method feature **** DONE ****
- **User Story 2:** As a developer, I want to familiarize myself with the transformative method feature **** DONE ****
- **User Story 3 & 4:** As a developer, I want to learn about the frameworks and technologies used in CyFinder **** DONE ****

Sprint 2
September 23rd – October 4th

Total Stories

- **User Story 5:** As a developer, I want to analyze the transformative method for undirectedness
- **User Story 6:** As a developer, I want to analyze the BFS algorithms from the source code
- **User Story 7:** As a developer, I want to analyze the DFS algorithm from the source code
- **User Story 8:** As a developer, I want to study the edge comparator class to determine if it can filter networks by weight
- **User Story 9:** As a developer, I want to test the additive vs transformative method for pre-processing code

Ariel Sari

- **User Story 6:** As a developer, I want to analyze the BFS algorithms from the source code
- **User Story 9:** As a developer, I want to test the additive vs transformative method for pre-processing code

Evyatar Saias

- **User Story 5:** As a developer, I want to analyze the transformative method
- **User Story 7:** As a developer, I want to analyze the DFS algorithm from the source code
- **User Story 8:** As a developer, I want to study the edge comparator class to determine if it can filter networks by weight

Sprint Review

Attendees: Ananda Mondal, Raihanul Tanvir, Ariel Sari, Evyatar Saias

Date: October 3rd, 2019

Start Time: 5:00pm

End Time: 7:00pm

The following stories were accepted after the approval of the PO during the sprint review:

- **User Story 6:** As a developer, I want to analyze the BFS algorithms from the source code **** DONE ****

- **User Story 9:** As a developer, I want to test the additive vs transformative method for pre-processing code **** DONE ****
- **User Story 5:** As a developer, I want to analyze the transformative method for undirectedness **** DONE ****
- **User Story 7:** As a developer, I want to analyze the DFS algorithm from the source code **** DONE ****
- **User Story 8:** As a developer, I want to study the edge comparator class to determine if it can filter networks by weight **** DONE ****

Sprint 3

October 7th – October 18th

Total Stories

- **User Story 10:** As a developer, I want to fix breadth first search algorithm code
- **User Story 11:** As a developer, I want to attach node attribute data to the networks in Cytoscape
- **User Story 12:** As a developer, I want to fix the .pom file for successful Maven integration with CyFinder
- **User Story 13:** As a developer, I want to create an interface for communication between the CyFinder module and subgraph finder module
- **User Story 14:** As a user, I want to know how to make my network undirected
- **User Story 15:** As a developer, I want to create a singleton class that would manage shared data between modules

Ariel Sari

- **User Story 11:** As a developer, I want to attach node attribute data to the networks in Cytoscape
- **User Story 12:** As a developer, I want to fix the .pom file for successful Maven integration with CyFinder
- **User Story 14:** As a user, I want to know how to make my network undirected

Evyatar Saias

- **User Story 10:** As a developer, I want to fix breadth first search algorithm code
- **User Story 13:** As a developer, I want to create an interface for communication between the CyFinder module and subgraph finder module

- **User Story 15:** As a developer, I want to create a singleton class that would manage shared data between modules

Sprint Review

Attendees: Ananda Mondal, Raihanul Tanvir, Ariel Sari, Evyatar Saias

Date: October 17th, 2019

Start Time: 5:00pm

End Time: 6:30pm

The following stories were accepted after the approval of the PO during the sprint review:

- **User Story 11:** As a developer, I want to attach node attribute data to the networks in Cytoscape **** DONE ****
- **User Story 12:** As a developer, I want to fix the .pom file for successful Maven integration with CyFinder **** DONE ****
- **User Story 14:** As a user, I want to know how to make my network undirected **** DONE ****
- **User Story 10:** As a developer, I want to fix breadth first search algorithm code **** DONE ****
- **User Story 13:** As a developer, I want to create an interface for communication between the CyFinder module and subgraph finder module **** DONE ****
- **User Story 15:** As a developer, I want to create a singleton class that would manage shared data between modules **** DONE ****

Sprint 4

October 21st – November 1st

Total Stories

- **User Story 16:** As a developer, I want to implement a function that reads the names of the genes from imported data
- **User Story 17:** As a developer, I want to implement a function that parses the imported data into the correct format
- **User Story 18:** As a developer, I want to implement a function that writes the data into the node table in Cytoscape
- **User Story 19:** As a user, I want to know how to use the bipartite condition on my network

- **User Story 20:** As a developer, I want to modify the node tables internal attributes to make it subgraph dependent
- **User Story 21:** As a developer, I want to generate a .jar file from the subgraph finder module which contains .class files along with its corresponding source code

Ariel Sari

- **User Story 18:** As a developer, I want to implement a function that writes the data into the node table in Cytoscape
- **User Story 20:** As a developer, I want to modify the node tables internal attributes to make it subgraph dependent
- **User Story 21:** As a developer, I want to generate a .jar file from the subgraph finder module which contains .class files along with its corresponding source code

Evyatar Saias

- **User Story 16:** As a developer, I want to implement a function that reads the names of the genes from imported data
- **User Story 17:** As a developer, I want to implement a function that parses the imported data into the correct format
- **User Story 19:** As a user, I want to know how to use the bipartite condition on my network

Sprint Review

Attendees: Ananda Mondal, Raihanul Tanvir, Ariel Sari, Evyatar Saias

Date: October 31st, 2019

Start Time: 5:00pm

End Time: 7:30pm

The following stories were accepted after the approval of the PO during the sprint review:

- **User Story 18:** As a developer, I want to implement a function that writes the data into the node table in Cytoscape **** DONE ****
- **User Story 20:** As a developer, I want to modify the node tables internal attributes to make it subgraph dependent **** DONE ****
- **User Story 21:** As a developer, I want to generate a .jar file from the subgraph finder module which contains .class files along with its corresponding source code **** DONE ****
- **User Story 16:** As a developer, I want to implement a function that reads the names of the genes from imported data **** DONE ****

- **User Story 17:** As a developer, I want to implement a function that parses the imported data into the correct format **** DONE ****
- **User Story 19:** As a user, I want to know how to use the bipartite condition on my network **** DONE ****

Sprint 5

November 4th – November 15th

Total Stories

- **User Story 22:** As a developer, I want Maven to include local dependencies in its build path
- **User Story 23:** As a developer, I want to generate a .jar file from the CyFinder module using Ant-builder
- **User Story 24:** As a developer, I want to a way of converting node objects into CyGrouperNode objects
- **User Story 25:** As a user, I want to know how to use the clique condition on my network
- **User Story 26:** As a developer, I want to integrate the CyGrouper package into the subgraph finder module
- **User Story 27:** As a developer, I want to convert the source-target relationship of genes into a Java Map object

Ariel Sari

- **User Story 25:** As a user, I want to know how to use the clique condition on my network **** DONE ****
- **User Story 26:** As a developer, I want to integrate the CyGrouper package into the subgraph finder module **** DONE ****
- **User Story 27:** As a developer, I want to convert the source-target relationship of genes into a Java Map object **** DONE ****

Evyatar Saias

- **User Story 22:** As a developer, I want Maven to include local dependencies in its build path **** DONE ****
- **User Story 23:** As a developer, I want to generate a .jar file from the CyFinder module using Ant-builder **** DONE ****

- **User Story 24:** As a developer, I want to a way of converting node objects into CyGrouperNode objects **** DONE ****

Sprint Review

Attendees: Ananda Mondal, Raihanul Tanvir, Ariel Sari, Evyatar Saias

Date: November 14th, 2019

Start Time: 5:00pm

End Time: 7:30pm

The following stories were accepted after the approval of the PO during the sprint review:

- **User Story 25:** As a user, I want to know how to use the clique condition on my network **** DONE ****
- **User Story 26:** As a developer, I want to integrate the CyGrouper package into the subgraph finder module **** DONE ****
- **User Story 27:** As a developer, I want to convert the source-target relationship of genes into a Java Map object **** DONE ****
- **User Story 22:** As a developer, I want Maven to include local dependencies in its build path **** DONE ****
- **User Story 23:** As a developer, I want to generate a .jar file from the CyFinder module using Ant-builder **** DONE ****
- **User Story 24:** As a developer, I want to a way of converting node objects into CyGrouperNode objects **** DONE ****

Sprint 6

November 18th – November 29th

Total Stories

- **User Story 28:** As a developer, I want to convert a Java Map object containing the names of the genes into a tree of CyGrouperNodes
- **User Story 29:** As a developer, I want to apply the breadth first search algorithm to group nodes into two distinct sets
- **User Story 30:** As a developer, I want to create a group column in the node table for each subgraph which contains each set a node belongs to
- **User Story 31:** As a developer, I want to map a Cytoscape GUI node into its tree entry
- **User Story 32:** As a user, I want to view which group a node belongs to in a bipartite subgraph

- **User Story 33:** As a user, I want to view which group a node belongs to in a clique subgraph
- **User Story 34:** As a developer, I want to create documentation for other developers to use

Ariel Sari

- **User Story 30:** As a developer, I want to create a group column in the node table for each subgraph which contains each set a node belongs to
- **User Story 31:** As a developer, I want to map a Cytoscape GUI node into its tree entry
- **User Story 32:** As a user, I want to view which group a node belongs to in a bipartite subgraph

Evyatar Saias

- **User Story 28:** As a developer, I want to convert a Java Map object containing the names of the genes into a tree of CyGrouperNodes
- **User Story 29:** As a developer, I want to apply the breadth first search algorithm to group nodes into two distinct sets
- **User Story 33:** As a user, I want to view which group a node belongs to in a clique subgraph
- **User Story 34:** As a developer, I want to create documentation for other developers to use

Sprint Review

Attendees: Ananda Mondal, Raihanul Tanvir, Ariel Sari, Evyatar Saias

Date: November 28th, 2019

Start Time: 5:00pm

End Time: 6:45pm

The following stories were accepted after the approval of the PO during the sprint review:

- **User Story 30:** As a developer, I want to create a group column in the node table for each subgraph which contains each set a node belongs to **** DONE ****
- **User Story 31:** As a developer, I want to map a Cytoscape GUI node into its tree entry **** DONE ****
- **User Story 32:** As a user, I want to view which group a node belongs to in a bipartite subgraph **** DONE ****
- **User Story 28:** As a developer, I want to convert a Java Map object containing the names of the genes into a tree of CyGrouperNodes **** DONE ****

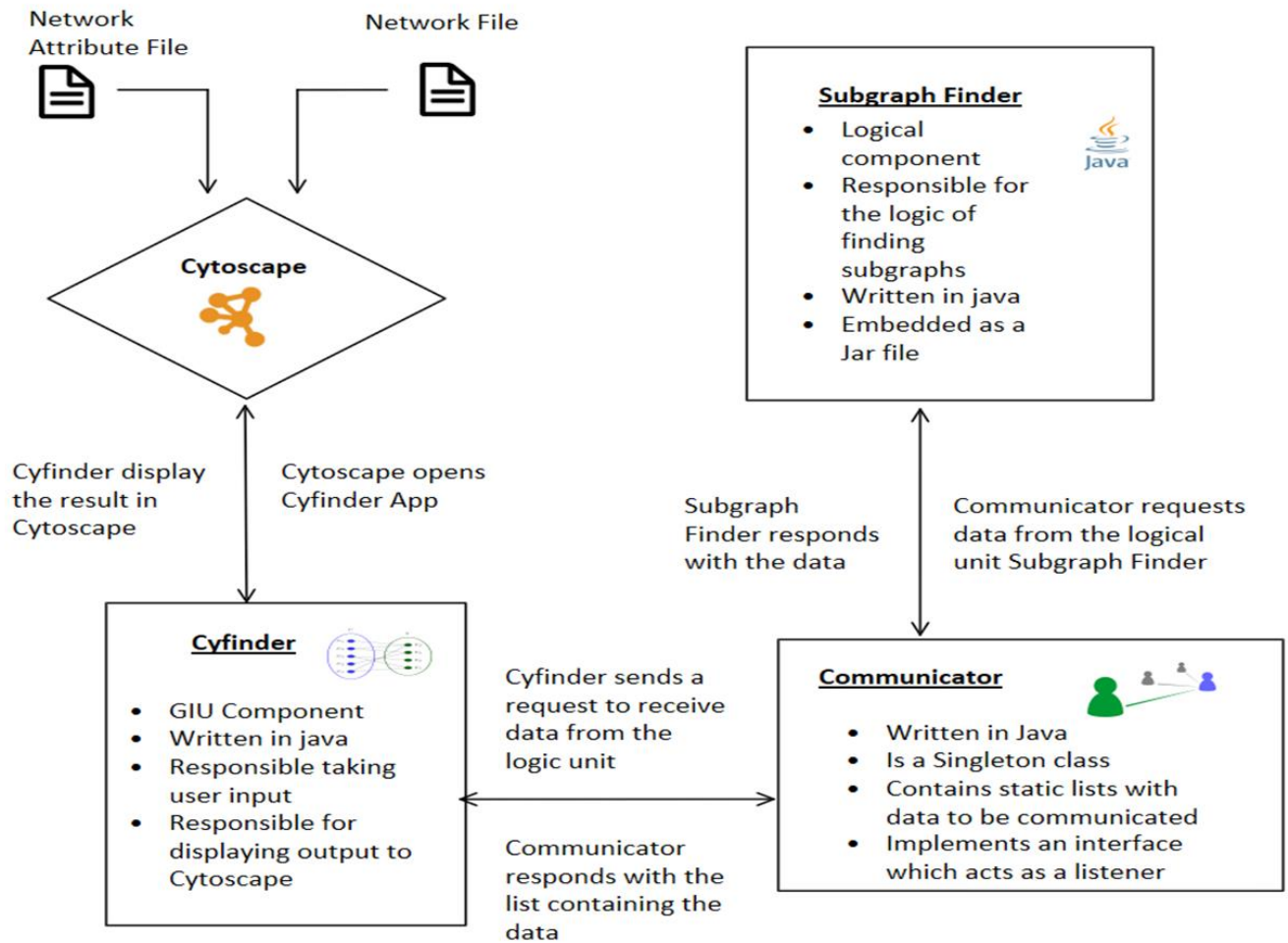
- **User Story 29:** As a developer, I want to apply the breadth first search algorithm to group nodes into two distinct sets **** DONE ****
- **User Story 33:** As a user, I want to view which group a node belongs to in a clique subgraph **** DONE ****
- **User Story 34:** As a developer, I want to create documentation for other developers to use **** DONE ****

SYSTEM DESIGN

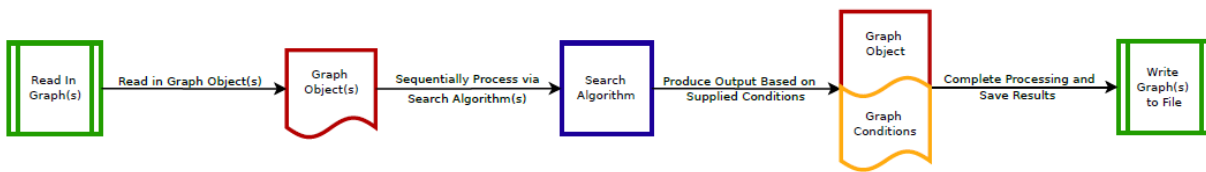
This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

System and Subsystem Decomposition

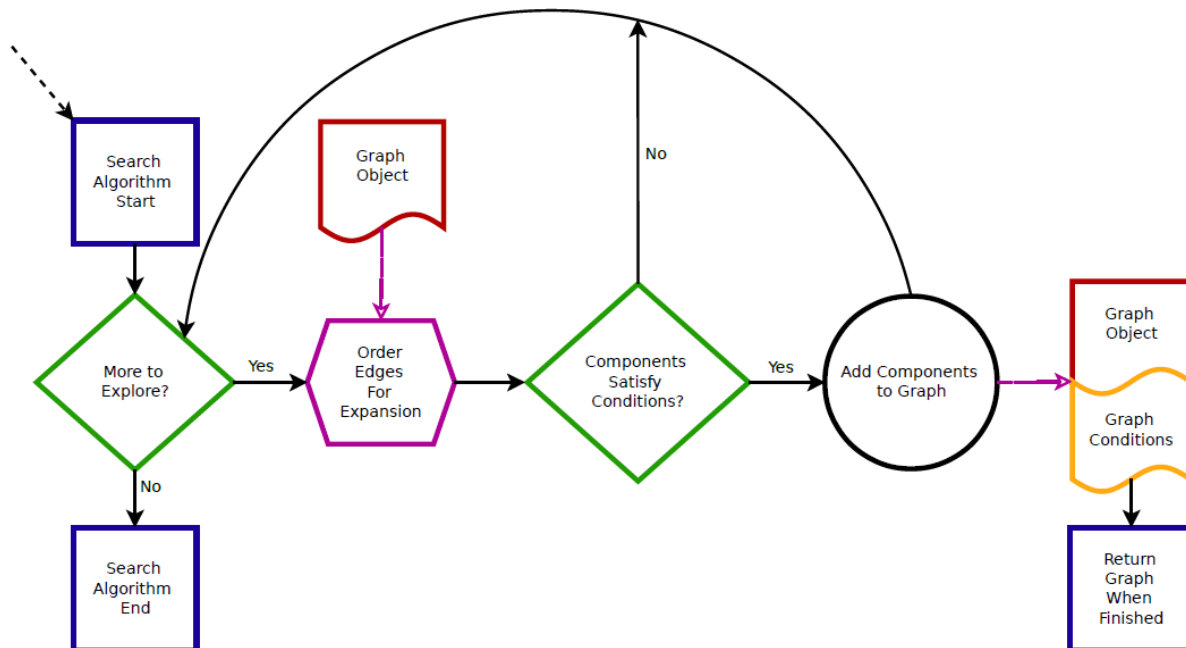
High level project overview:



Subgraph Finder General Process

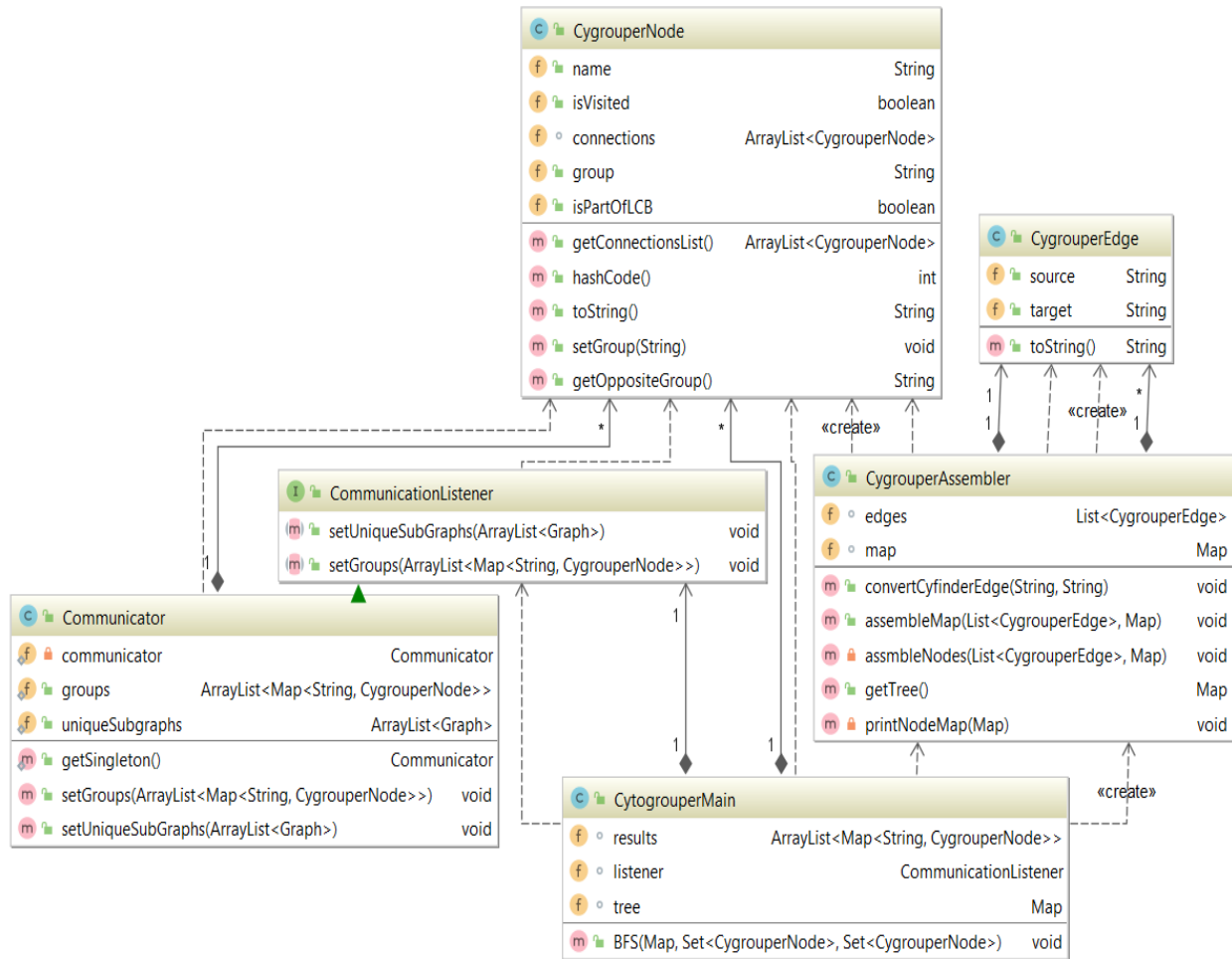


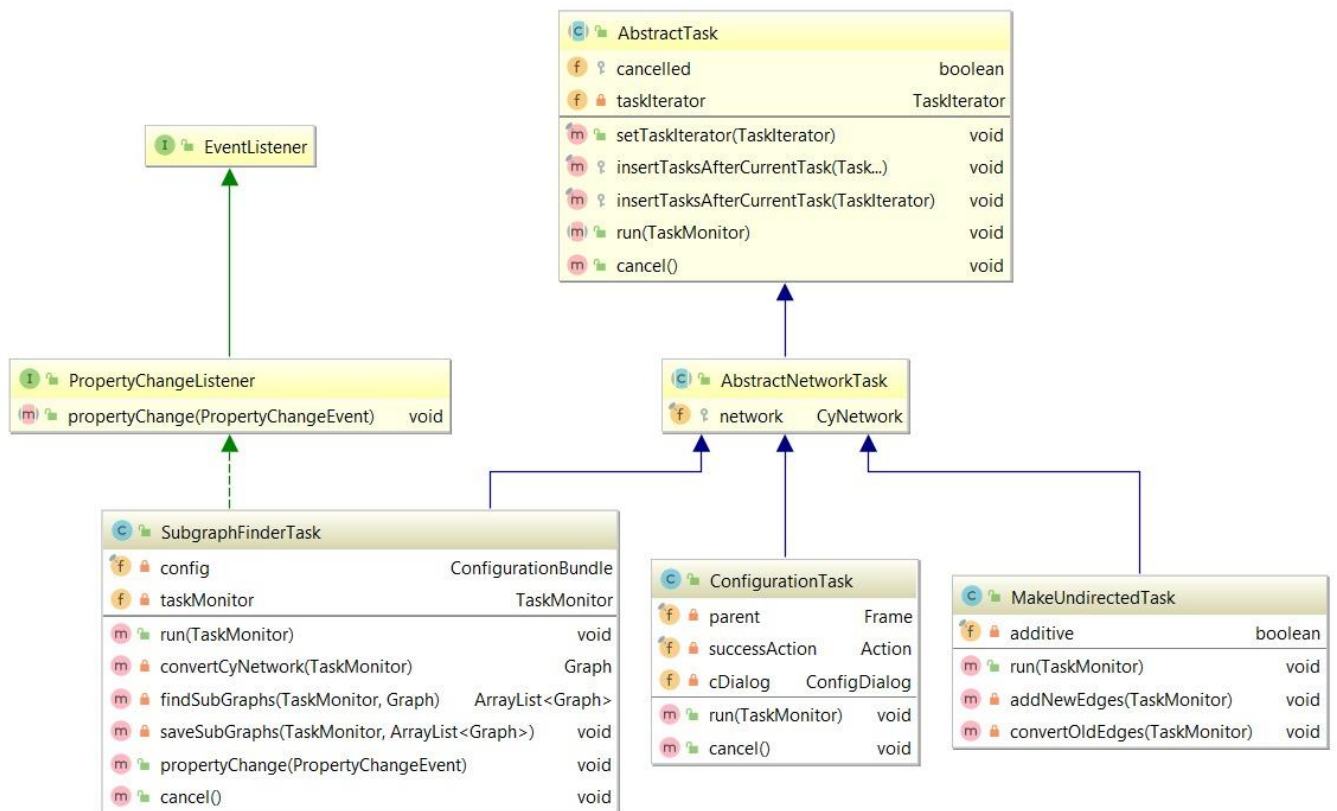
Constructive (Search-Based) Algorithm Process



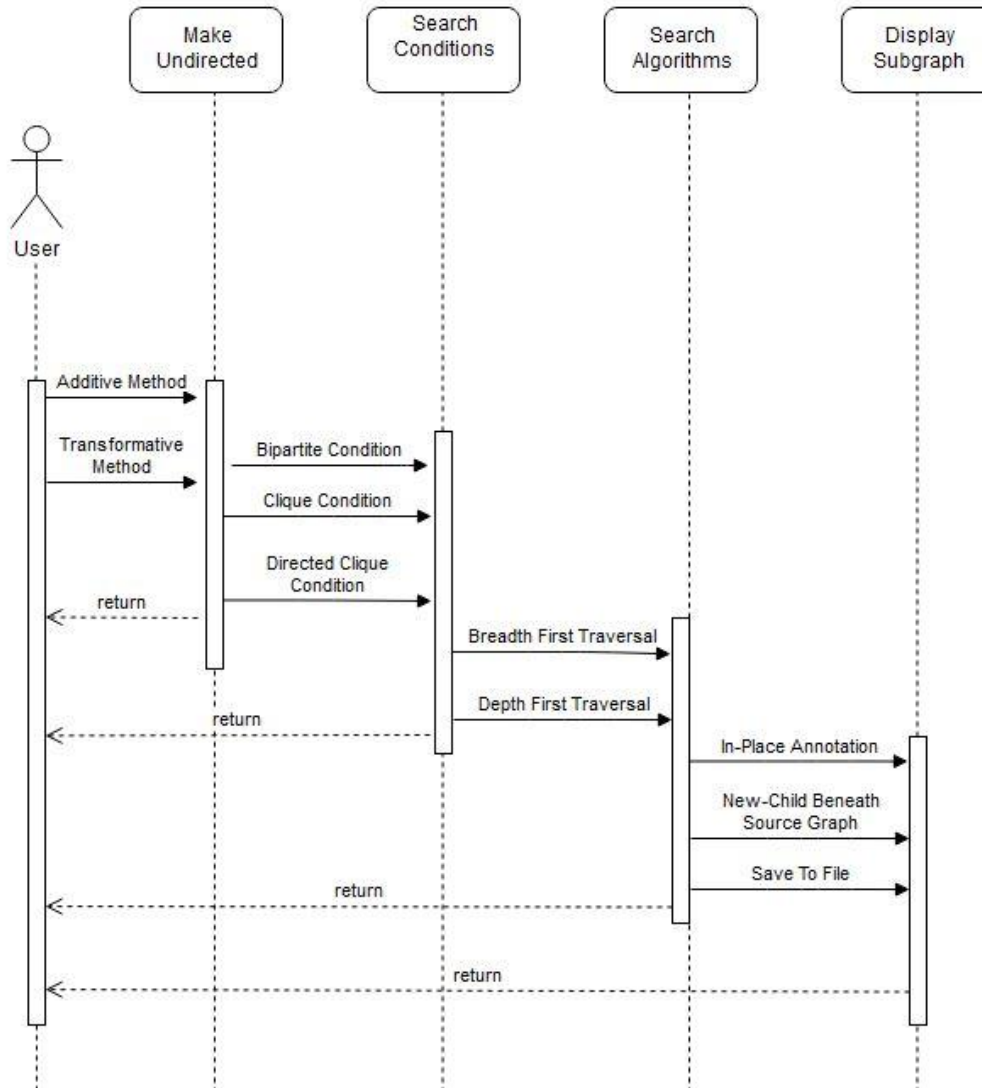
APPENDIX

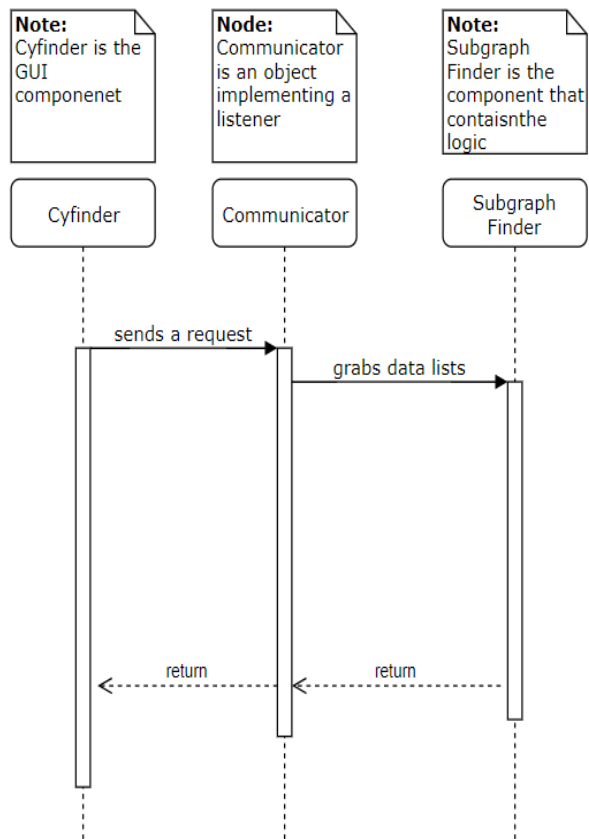
Appendix A - UML Diagrams



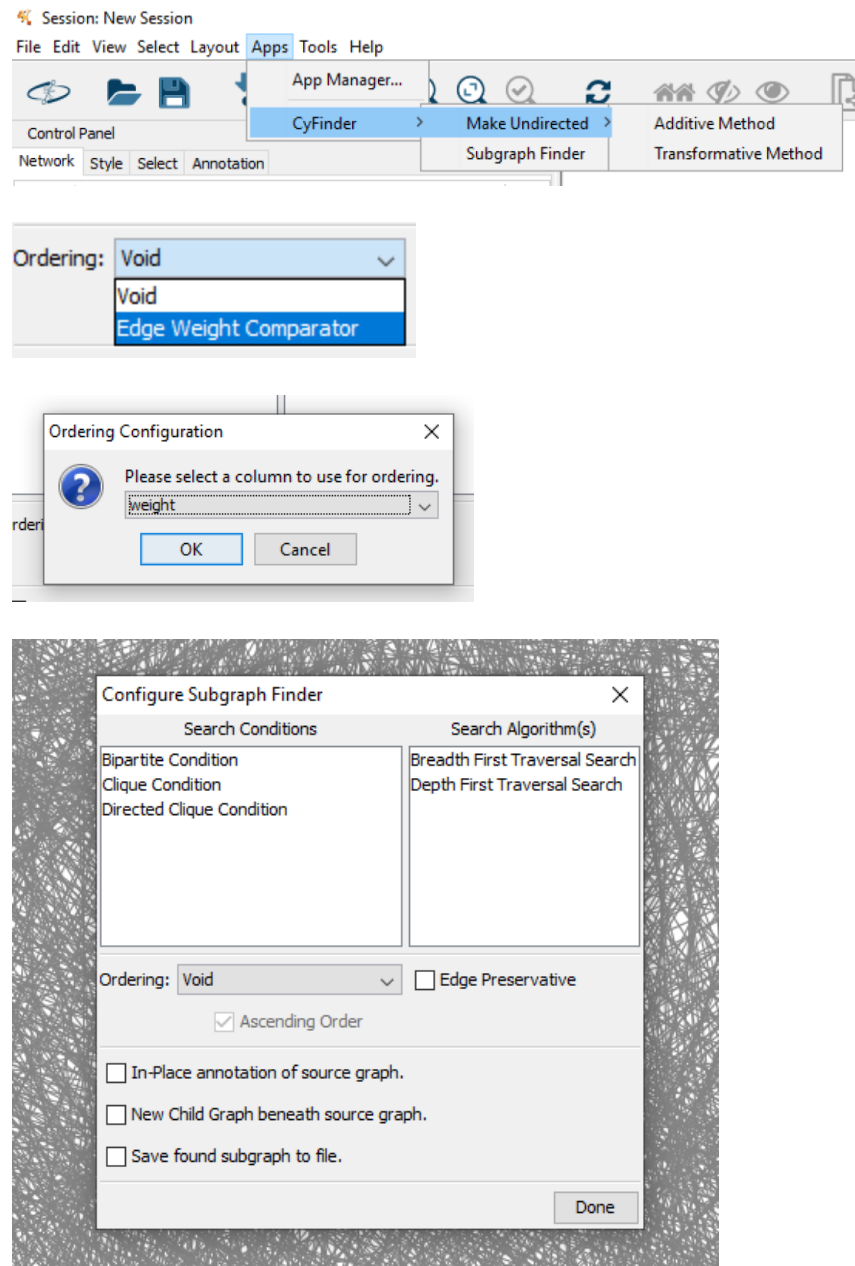


Appendix B - Sequence Diagrams





Appendix C - User Interface Design



Appendix E - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

Written by Ariel Sari & Evyatar Saia
Cytoscape 3.7.2

CyFinder User Guide

About

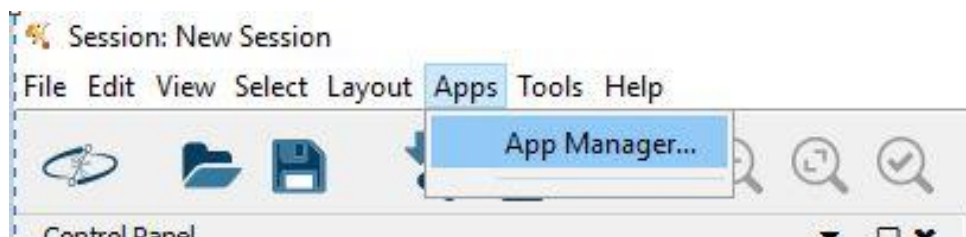
CyFinder is a Cytoscape app used for finding clique (complete) and bipartite subgraphs within a given network, which are then displayed in a visually meaningful way.

Getting Started

Step 1:

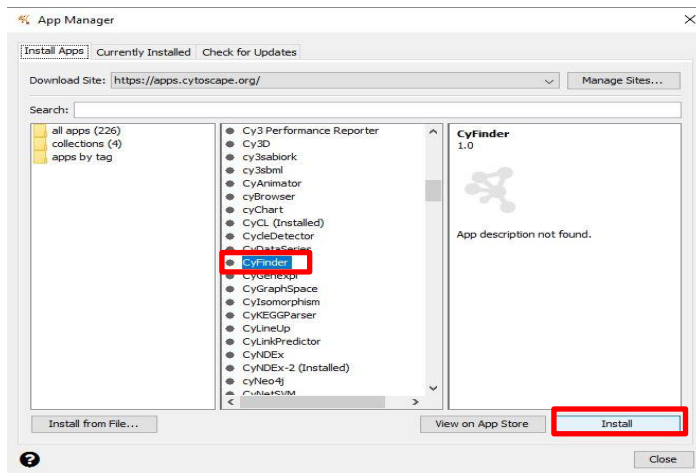
To get started we need to download the CyFinder app first.

To download CyFinder click the Apps toolbar and select the “App Manager...” option



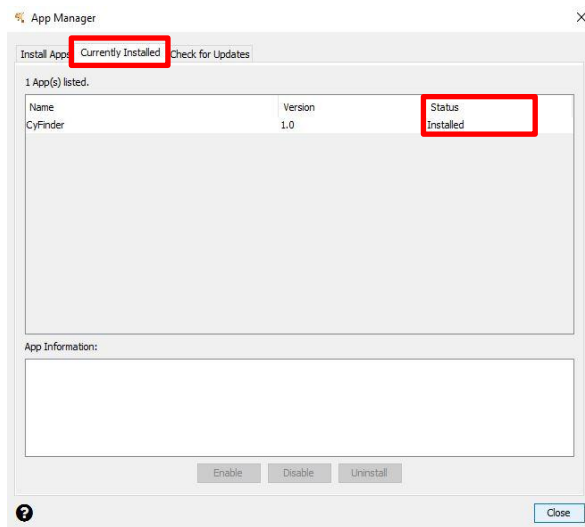
Step 2:

Next, locate the CyFinder app from the list of available apps and click install on the bottom right.

**Step 3:**

After clicking “Install”, click on the “Currently Installed” tab and a screen like this should appear.

If CyFinder has been installed successfully, you should see the word “Installed” under the “Status” column.

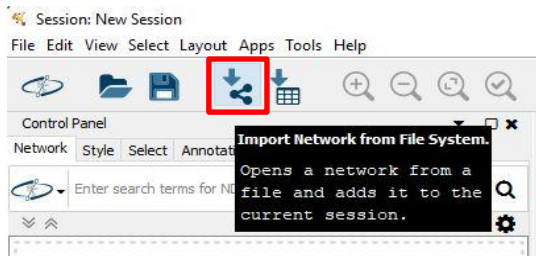


You may now close this window.

Step 4:

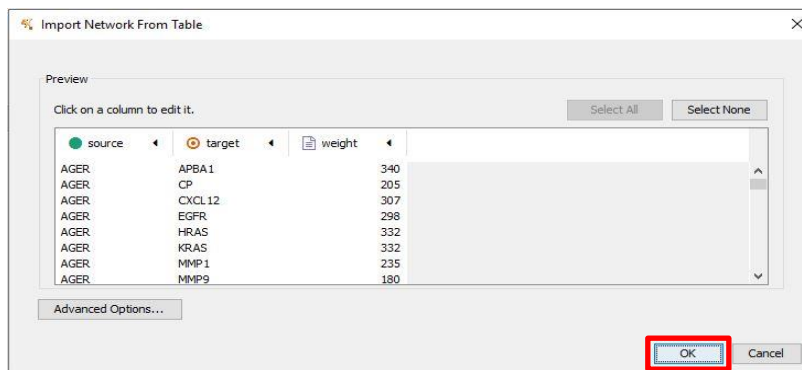
Now that we have CyFinder installed we are ready to import a network.

To import a network, click on the “Import Network from File System” button as shown below and navigate to the location of your network file then select it.

**Step 5:**

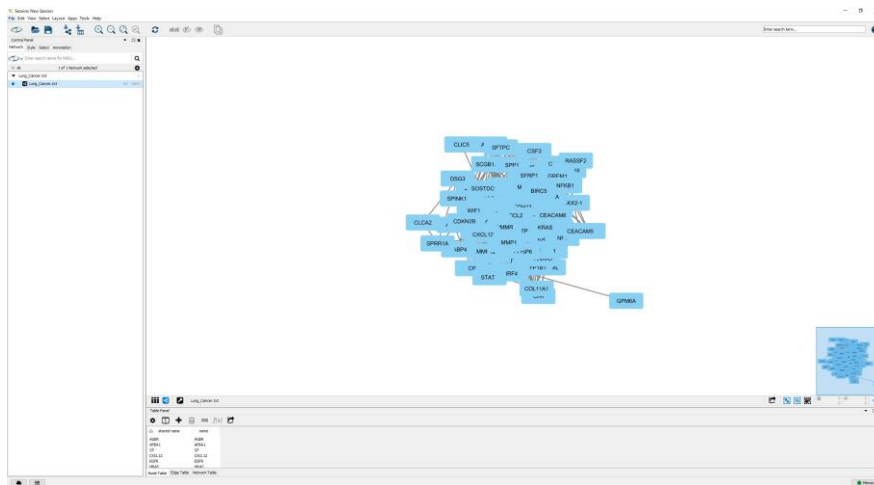
After selecting your network file from the previous step, a window like this should appear.

Make sure all your data has been processed successfully by Cytoscape and press “Ok” to import your network.



Step 6:

Now that you have imported your network it should look somewhat like the figure below. In this example we are using “The Lung_Cancer.txt” data set which makes up our network. This particular network contains 82 nodes and 1062 edges.



Now that you have your network imported into Cytoscape, we are ready to explore the options that the CyFinder app has to offer.

Step 7:

First, in order to find subgraphs within your network using the CyFinder app you will have to make your network undirected.

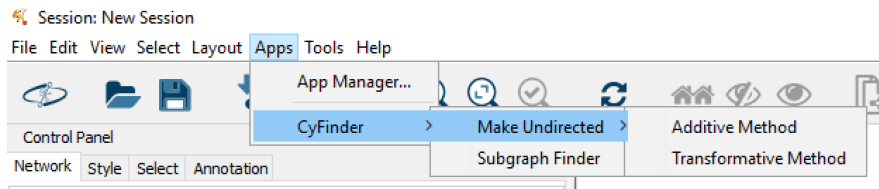
CyFinder provides two different options of doing this, the **additive method** and **transformative method**.

The additive method makes the network undirected by making a copy of every edge and reversing its source-target relationship.

The transformative method makes the network undirected by altering the existing edges and replacing them with exact duplicates except they have the internal Cytoscape flag for directedness of edges set to false.

In this tutorial, we will be using the Transformative Method.

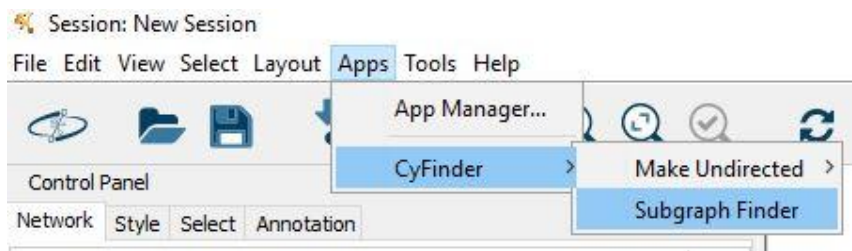
Click Apps -> CyFinder -> Make Undirected -> Transformative Method or Additive Method



Step 8:

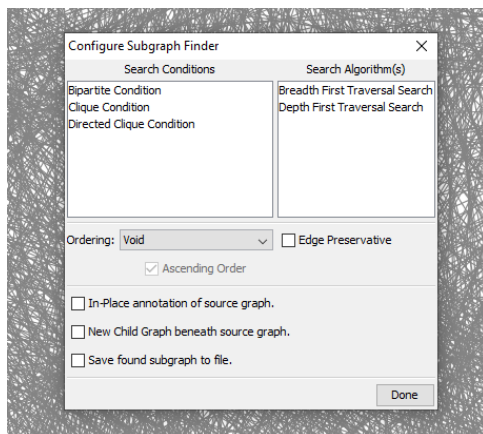
Now that our network is in an undirected state, we are ready to find meaningful subgraphs within this network.

Click on Apps -> CyFinder -> Subgraph Finder



Step 9:

After performing the previous step, the following window should appear:



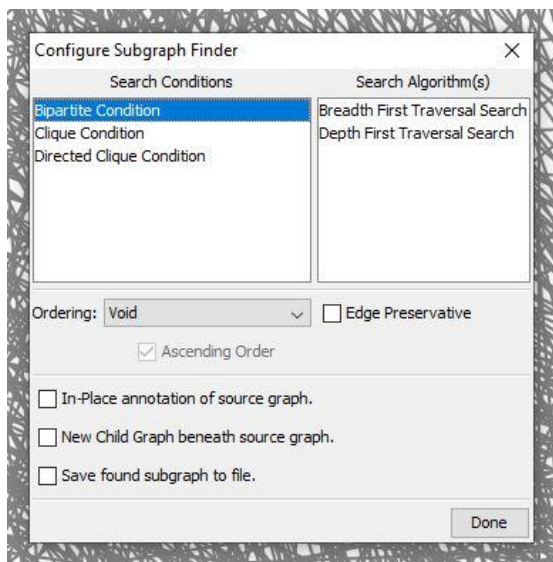
We are now ready to find Clique and Bipartite subgraphs.

Step 10:

As you can see CyFinder provides three search conditions to choose from, the **bipartite condition**, **clique condition**, and **directed clique condition**.

In this tutorial we will be showing the bipartite condition which is used to find bipartite subgraphs within the network, however the steps of using the clique condition and directed clique condition will be identical.

Select the **bipartite condition** by clicking on it.

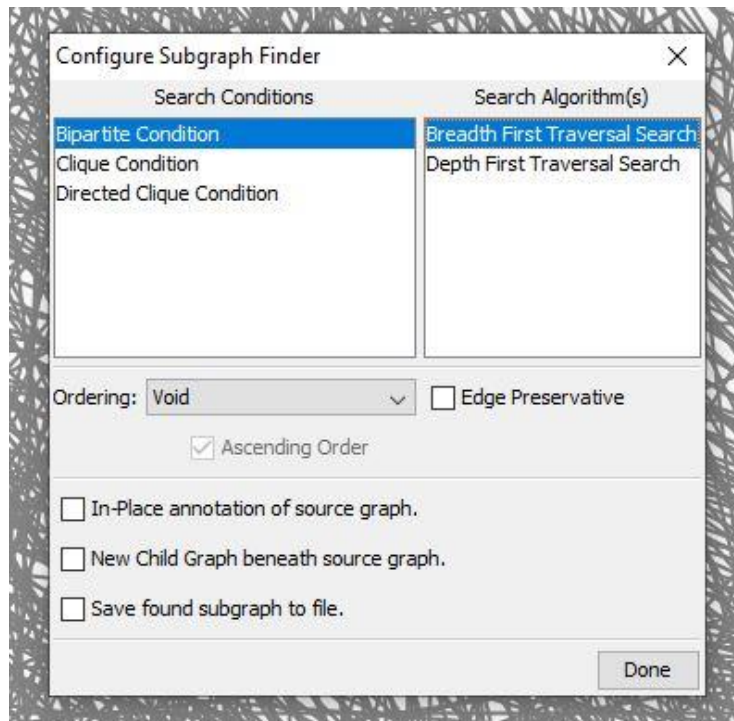
**Step 11:**

Now that we have specified which type of subgraphs should be searched for, we need to specify which search algorithm should be used to find them.

CyFinder provides us with two options to choose from, **breadth first traversal search**, and **depth first traversal search**.

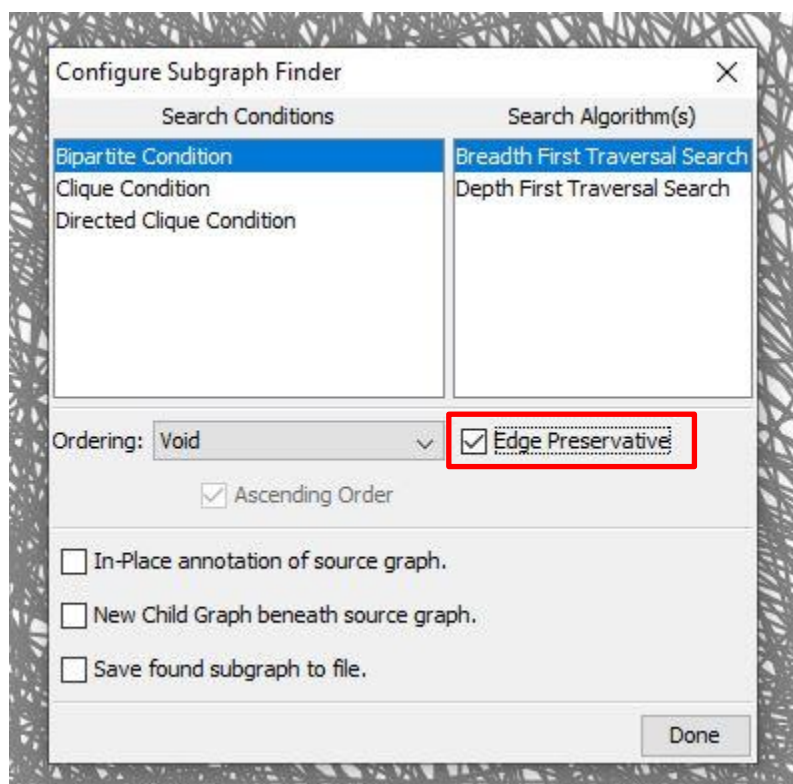
In this tutorial we will be using the breadth first traversal search algorithm, however the steps of using the depth first traversal search algorithm will be identical.

Select the **breadth first traversal search** algorithm by clicking on it.

**Step 12:**

Now that we have chosen our search condition and search algorithm, make sure the “**Edge Preservative**” checkbox is checked in order to preserve our edges.

Check the “**Edge Preservative**” checkbox by clicking on it.



Notice that there is a dropdown box for “Ordering” that we haven’t discussed yet that is currently is set to void.

We will leave it as void for the purposes of this tutorial since we are not interested in any particular ordering of our subgraphs.

Step 13:

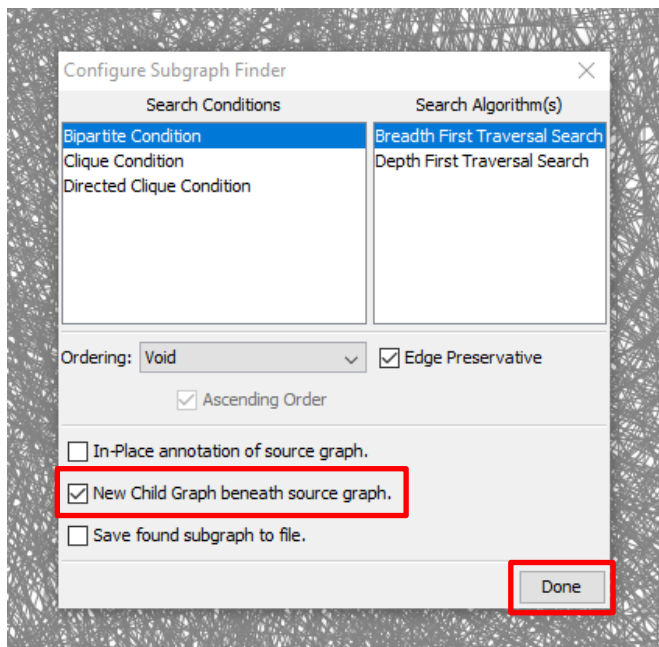
Now it is time to tell CyFinder how we want our results to be given to us. CyFinder provides us with three different options for how we would like to receive/display our results.

The three options are “**In-Place annotation of source graph**”, “**New Child Graph beneath source graph**”, and “**Save found subgraph to file**”, which we will now discuss.

In this tutorial we will be using the “New Child Graph beneath source graph”.

Check the “**New Child Graph beneath source graph**” checkbox by clicking on it.

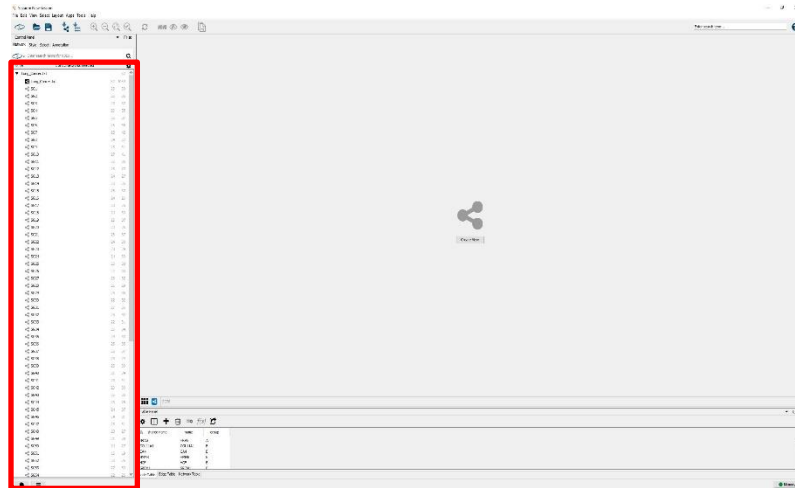
Now that all of your options selected click the “Done” button.



Step 14:

After you click on the “Done” button a new window will appear on left side of your Cytoscape window that will contain our results, as shown in the figure below.

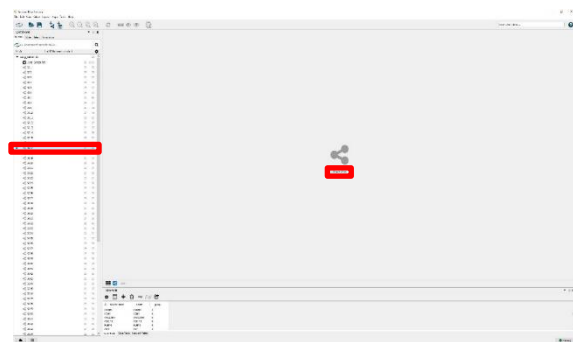
The results are all of the subgraphs that were found given the conditions that we specified. In our case, all of the subgraphs that were found are bipartite graphs using the breadth first search algorithm.

**Step 15:**

Now that we have all of our bipartite subgraphs available to us on the left window, we can simply select one.

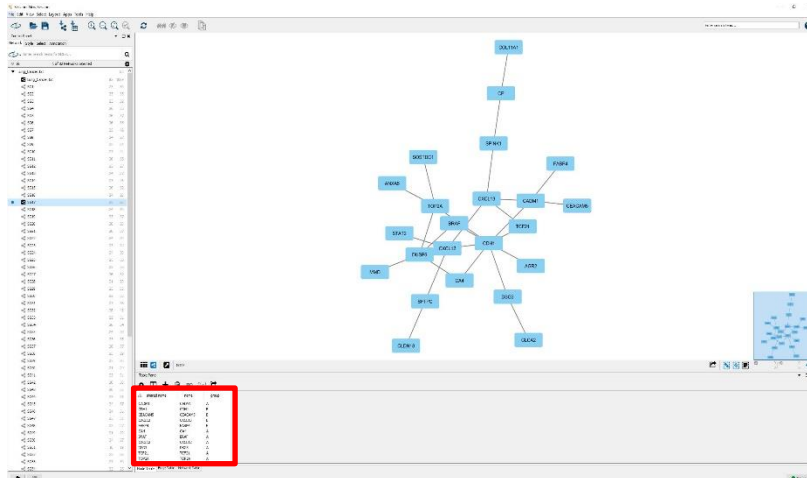
Then we can click on the “Create View” button in the center of the screen in order to view the selected subgraph.

In this tutorial we selected the “SG17” subgraph which stands for subgraph 17.

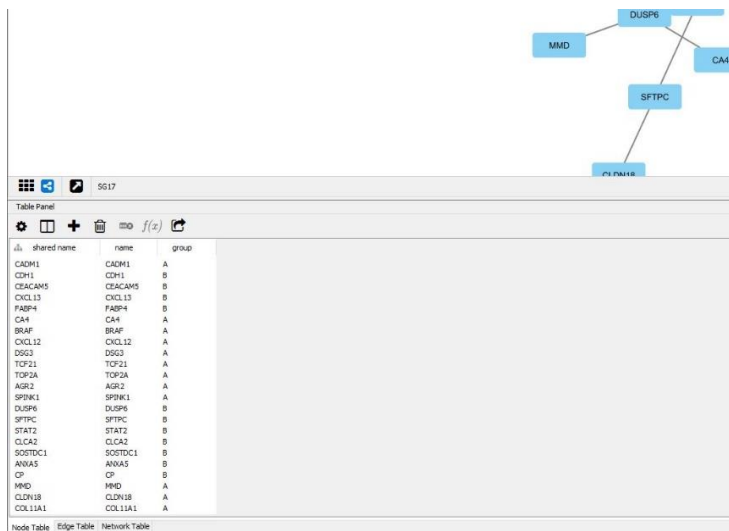


Step 16:

After clicking the “**Create View**” button your subgraph should be displayed. Notice the “Node Table” on the bottom of the window that CyFinder has populated particularly for this subgraph.

**Step 17:**

Still looking at the same subgraph “SG17” from our previous step. We can notice that it has a column called “group” in its node table. CyFinder has populated this column for us so we can easily be able to distinguish between the two different sets that make up this bipartite subgraph.



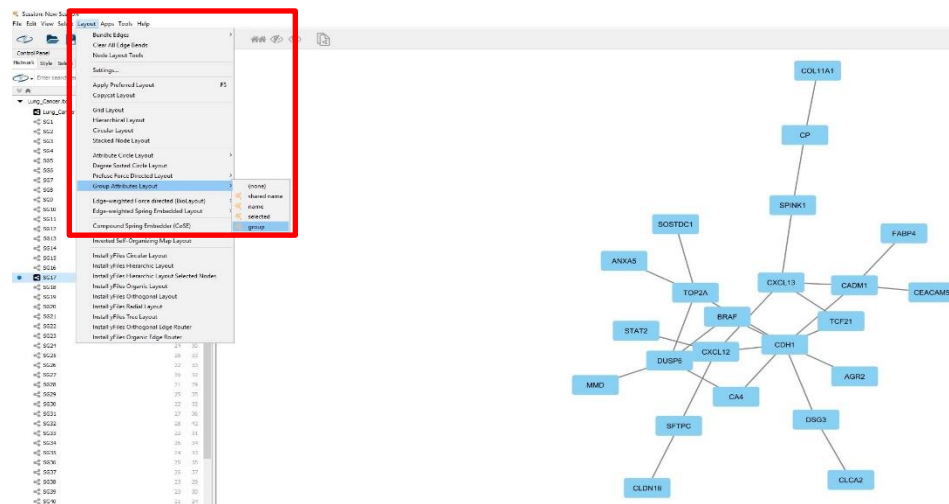
In the next step, we will show how to use the “group” attribute that CyFinder has provided to us in order to visually display the two distinct sets using the Cytoscape layout options.

Step 18:

To visually show the distinct sets using the “group” attribute that CyFinder has provided to us with, we can take advantage of the “Group Attribute Layout” provided in Cytoscape by default.

As show in the figure below,

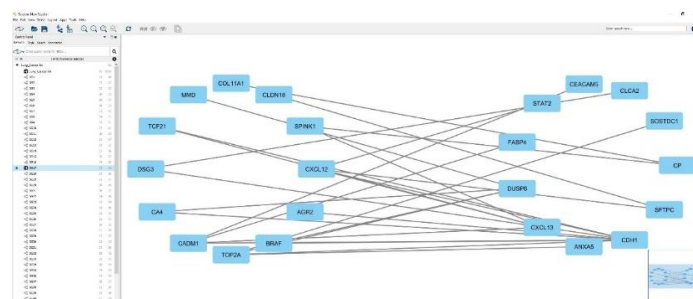
Click Layout -> Group Attribute Layout -> group



Step 19:

After performing the previous step correctly your bipartite subgraph should now be displayed in a similar format as shown in the figure below.

Now you should easily be able to distinguish between the two sets that make up the bipartite subgraph, where each distinct set is represented using a circular layout.



Written by Ariel Sari & Evyatar Saia
Cytoscape 3.7.2 and the Eclipse IDE

CyFinder Developer Guide

The CyFinder plugin is made up of two different modules, the “CyFinder” module and the “Subgraph Finder” module. Both modules should be treated as two separate projects.

The CyFinder module is responsible for the GUI of the plugin on Cytoscape while the Subgraph Finder module is responsible for the logic of the plugin.

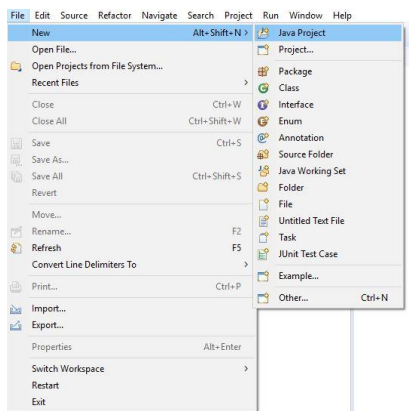
The way this plugin works is by having the Subgraph Finder module as a jar file inside the CyFinder jar which is generated from the CyFinder project using Maven. Basically, it is a jar inside a jar, a bit confusing but hopefully the following images and step by step explanation will help.

How to develop

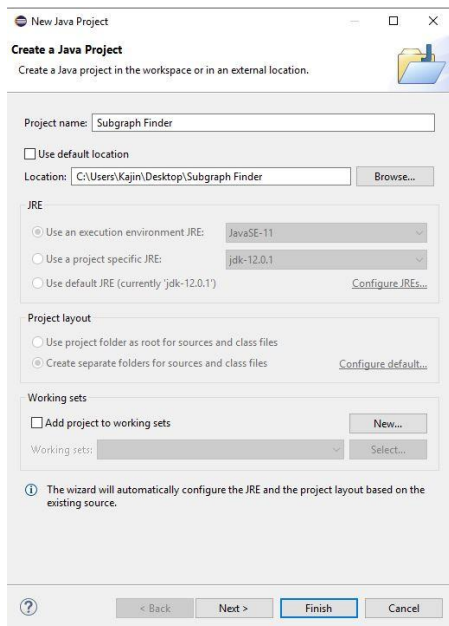
First download the CyFinder and Subgraph Finder projects from the Google Drive folder that should be shared with you by your product owner and import them as two individual projects into your favorite IDE. All the following steps shown in this tutorial will be done in the Eclipse IDE but it should work with any IDE.

We will first import the “Subgraph Finder” project into Eclipse

Step 1: Select file in the toolbar and choose the New -> Java Project option from the dropdown.

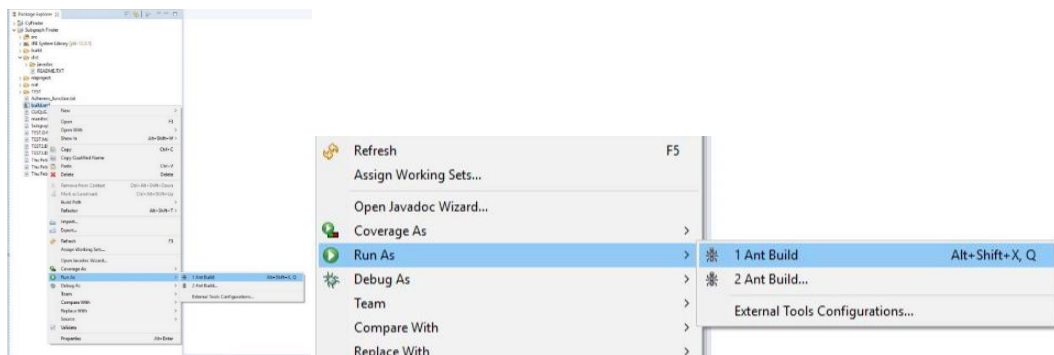


Step 2: Uncheck the box “Use default location” and click the “Browse...” button. Then choose the directory of the Subgraph Finder project after you download and extract it from the shared Google Drive folder and click Finish.



Now that the “Subgraph Finder” project is imported into your IDE; we will generate a “SubgraphFinder.jar” file that will be used as a dependency by the “CyFinder” project. This will be shown in later steps.

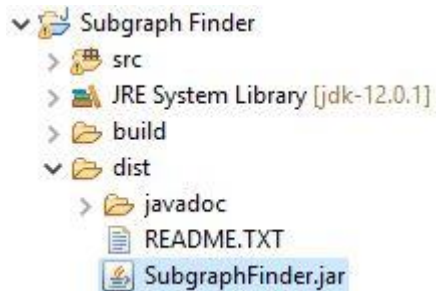
Step 3: Right click on the “build.xml” file in the Subgraph Finder project and choose Run As -> Ant Build



An example of how a successful ant build should look like:

```
Console
<terminated> Subgraph Finder build.xml [Ant Build] C:\Program Files\Java\jdk-12.0.1\bin\javaw.exe (Nov 19, 2019, 11:28:53 PM)
-do-jar-delete-manifest:
-do-jar-without-libraries:
-do-jar-with-libraries:
-post-jar:
-do-jar:
jar:
-javadoc-build:
-javadoc-browse:
javadoc:
default:
BUILD SUCCESSFUL
Total time: 816 milliseconds
```

Step 4: After a successful Ant Build, the jar file that was generated called “SubgraphFinder.jar” should be found in the “dist” folder in your “Subgraph Finder” project.



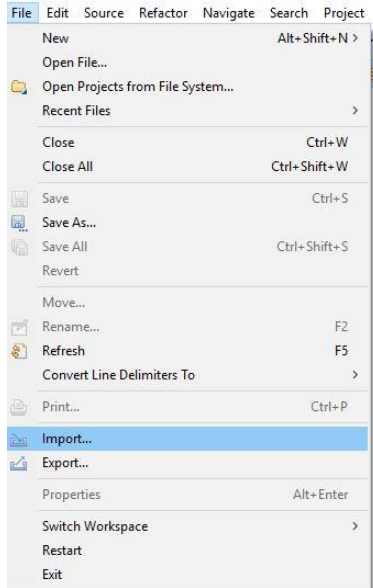
```
Subgraph Finder
├── src
├── JRE System Library [jdk-12.0.1]
├── build
├── dist
│   ├── javadoc
│   ├── README.TXT
│   └── SubgraphFinder.jar
```

NOTE: Everytime you make changes to anything in the “SubgraphFinder” project which will usually be logical changes you will have to generate a new jar file in this manner. This jar file is then used as a dependency for the CyFinder project.

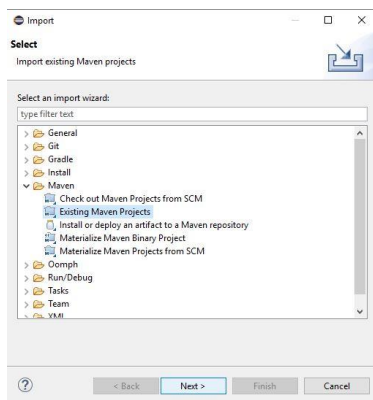
We will now import the CyFinder project. This is a Maven project and should be imported as such with the following steps:

Note: If Maven is not already installed as part of your IDE package follow the link below for instructions on how to install and run Maven <https://maven.apache.org/index.html>

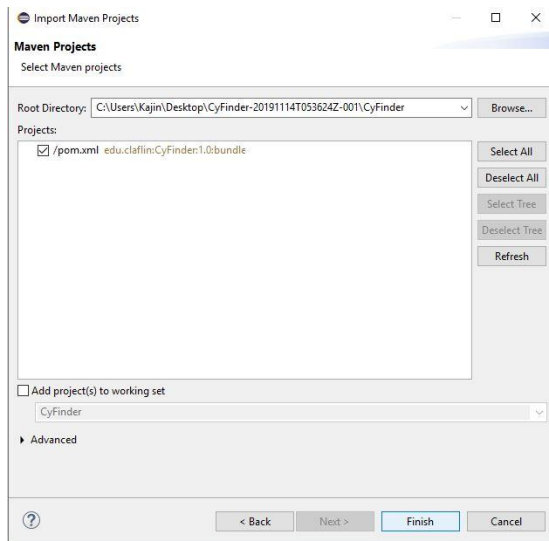
Step 5: Select file in the toolbar and choose the Import option from the dropdown.



Step 6: Select Maven and then choose the option “Existing Maven Projects”



Step 7: Click the “Browse...” button and then choose the directory of the CyFinder project after you download and extract it from the Google Drive shared folder and click “Finish”.



After importing the “CyFinder” project you will notice an error in the “pom.xml” file that has to do with the dependency “edu.claflin:finder:jar:3.0”.

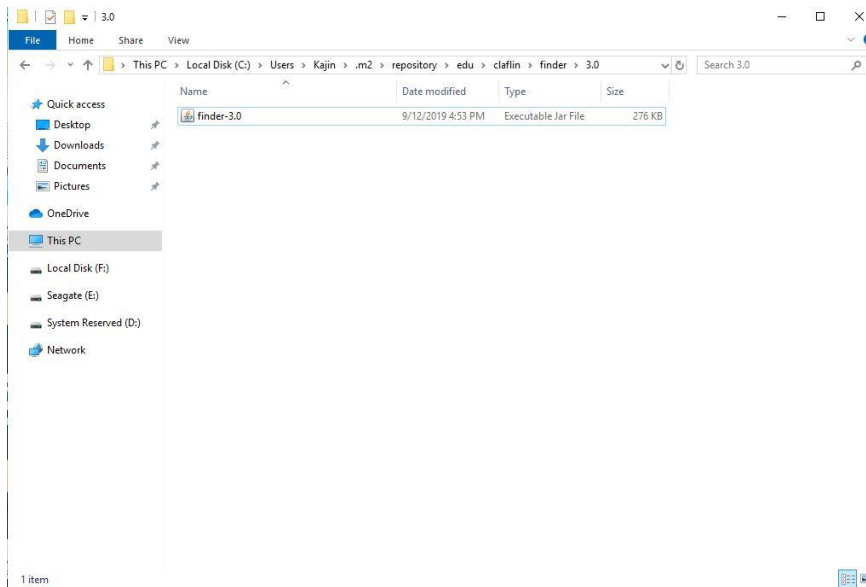


This error is basically stating that this particular dependency can’t be found by Maven and should be installed manually for CyFinder to work. This dependency is actually the jar file called “SubgraphFinder.jar” that we created in the previous steps using the Subgraph Finder project. As mentioned earlier, this jar can be found in the “dist” folder in the Subgraph finder project if you followed the previous steps correctly.

Note: This file is also available on Google Drive with the name “finder-3.0.jar”. However, if you made any changes to the Subgraph Finder project that you would like to keep. You will have to generate a new jar file using the previous steps each time.

Step 8: In order to use this jar file as a Maven dependency for the CyFinder project you must first rename the jar from “SubgraphFinder.jar” to “finder-3.0” and then place the jar file into the following directory:

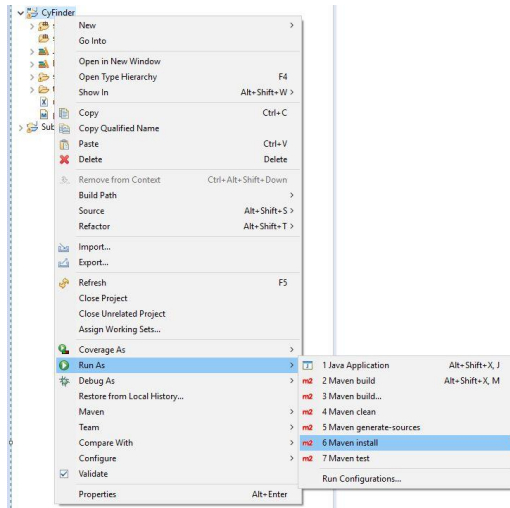
C:\Users\<YourUserNameHere>\.m2\repository\edu\claflin\finder\3.0



Now you should notice that error regarding the dependency “edu.claflin:finder:jar:3.0” in the CyFinder project from before has been cleared and you should have no other errors.

Now to test that everything works we will generate a jar file from the CyFinder project using **Maven** that we will then import into Cytoscape.

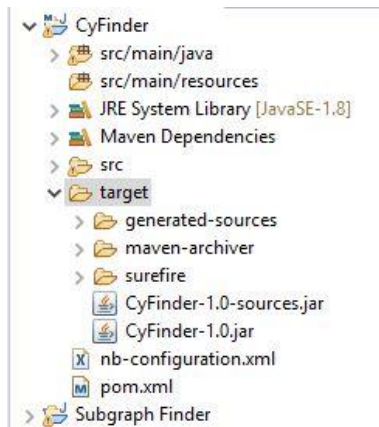
Step 9: Right click the CyFinder project and choose the options Run As -> Maven install from the dropdown menu



If done correctly the output below should be displayed in the console

```
[INFO] <<< maven-bundle-plugin:2.5.4:install (default-install) < install @ CyFinder <<<
[INFO]
[INFO]
[INFO] --- maven-bundle-plugin:2.5.4:install (default-install) @ CyFinder ---
[INFO] Installing edu/claflyn/CyFinder/1.0/CyFinder-1.0.jar
[INFO] Writing OBR metadata
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.410 s
[INFO] Finished at: 2019-11-19T22:34:48-05:00
[INFO] -----
```

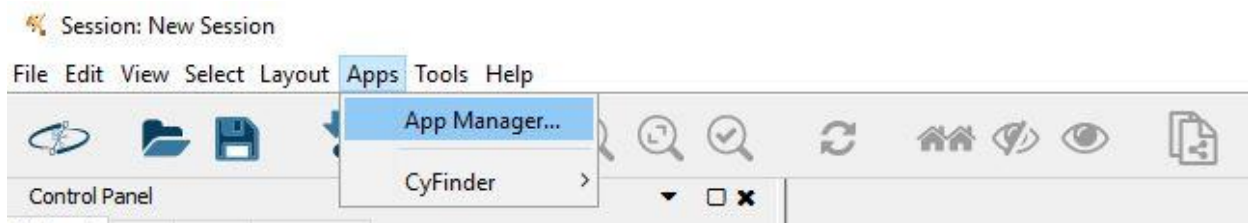
A successful Maven install should generate a target folder in your project's directory. This folder will contain two jar files, the one that we will use for Cytoscape is the “CyFinder-1.0.jar” file.



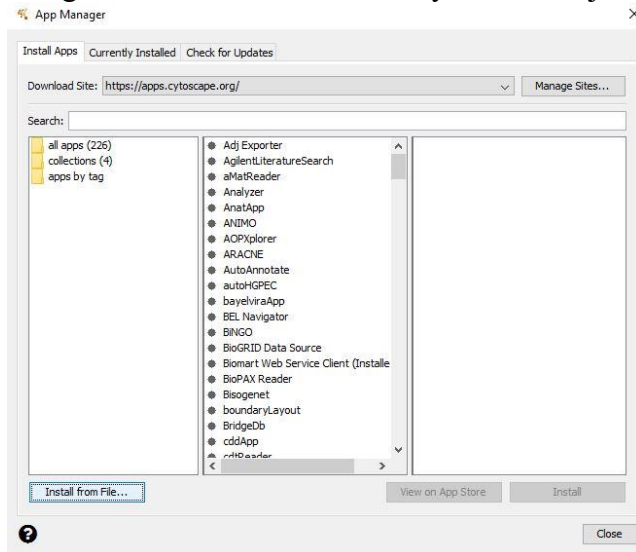
Note: Everytime you do a Maven install the old target folder will be deleted and a new one will be generated.

We will now import the “CyFinder-1.0.jar” file that we generated into Cytoscape.

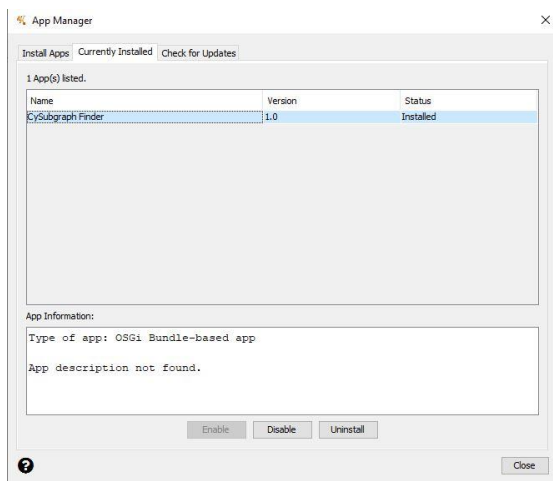
Step 10: Open Cytoscape and select Apps from the toolbar then click on “App Manager”



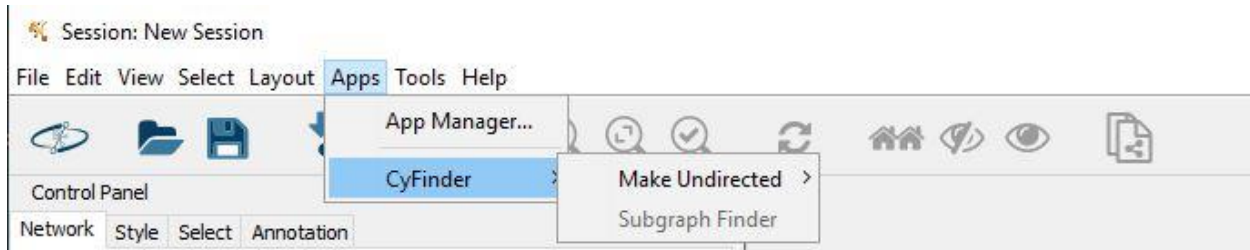
Step 11: Click on the “Install from File...” button on the bottom left of the window. Then navigate to the location of the “CyFinder-1.0.jar” which should still be inside your target folder.



Step 12: If no errors have occurred you should see the following window after selecting your jar file. The indication of success is if the word under Status says “Installed”. You can now close this window.



Step 13: Lastly, confirm that you now have access to CyFinder. You should be able to see the CyFinder option from the Apps toolbar dropdown menu.



IMPORTANT NOTES:

- For testing changes made to the **CyFinder** Project (GUI): You will have to only generate a new CyFinder jar using maven and then import it to Cytoscape.
- For testing changes made to the **Subgraph Finder** Project (Logic): you will have to go through all the steps of generating both jar files and then ensuring that you replace the old jar generated by Subgraph finder with the new, renaming it each time. Then importing the new CyFinder jar generated with Maven into Cytoscape to test your code.
- If any errors regarding the JavaDocs occur from either the CyFinder project or Subgraph Finder project, you can simply comment out any reference to the JavaDocs from the project giving you the errors.
- Developer information for Cytoscape available at this link https://cytoscape.org/documentation_developers.html and the Cytoscape API JavaDocs can be found here <http://chianti.ucsd.edu/cytoscape-3.6.0/API/>

Wishlist

- More sorting algorithms
Ex. average, descending order
- Finding largest complete bipartite subgraph within a generated bipartite subgraph.
- Assign a unique set id to each level of the bipartite tree generated by the bipartite condition and map the tree nodes to their corresponding set based on which level they belong to within the tree.
- Create custom layouts to display the generated subgraphs in a way that provides more meaning out of the data displayed.
- Improve UI for more intuitive user experience.
- Add a description and image on the Cytoscape App store for the CyFinder app which better represents what that app currently does.