

Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title: App Development for Analyzing Biological
Networks– CyFinder

Team Members: David Barahona, Gabriel Zavala, Riguens Brutus

Product Owner(s): Ananda Mondal

Mentor(s): Ananda 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

This document presents the information necessary to gain a good understanding of the functionality and work done on the Cytoscape plug-in or app called CyFinder. The primary purpose of the CyFinder is to process biological networks that are imported in the form of graphs where each node represents a protein or gene, and each edge represents the interactivity of the objects. This project focuses on the implementation directed weighted networks, improvement, and enhancement of the algorithms plus the addition of new algorithms used by the Infomap community detection algorithm. This algorithm can be used to detect clusters formed by biological entities to detect various diseases like cancer.

APPENDIX

1 Introduction.....	4
2 Current System.....	5
3 Purpose of New System	5
4 User Stories	7
5 Implemented User Stories.....	7
6 Pending User Stories.....	7
7 Project Plan	8
8 Hardware and Software Resources	8
9 Sprint Plans	9
10 System Design	16
11 Project Dependencies Diagram	17
12 Design Patterns	17
13 Sprint Review Reports	18
14 User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document, and other documents	25
14.1 How to install CyFinder	25
14.2 How to develop	25
14.3 Importing Subgraph Finder:	26
14.4 Importing and Building CyFinder:.....	26
14.5 Developer Notes	27
14.6 User Guide.....	28
14.7 Wishlist.....	29
15 References.....	29

1 Introduction

Cytoscape is an open-source tool designed for network analysis and visualization. It is used in various fields, including molecular network and gene expression analysis. CySubgraphFinder, also referred to as CyFinder, is an extension for Cytoscape. This plugin enhances Cytoscape's existing network analysis and visualization capabilities. CyFinder is distributed as a Java Archive (JAR) file, which is installed using Maven.

Most of CyFinder's functionality resides in SubgraphFinder, which encompasses the logic for all the algorithms integrated into Cytoscape by CyFinder. This project aims to enhance and improve the existing implementation of the Infomap algorithm by refining the current algorithm and adding new ones. Additionally, we have improved the community detection feature by introducing support for a Directed Weighted Networks Implementation in Map equation and Infomap.

2 Current System

The current system enables users to convert directed graphs to undirected graphs using either an additive approach, which adds another edge between connected nodes, or a transformative approach, altering the edge object for CyFinder compatibility. It allows users to specify search conditions (bi-partite, clique, or directed clique), select the search algorithm (Breadth or Depth First Search), decide if the graphs should be ordered (based on edge weight and SUID), and choose the method for saving generated subgraphs (adding a truth table, generating separate subgraphs). Users can also sort graphs by node count, visualize specific k-partite graphs based on input, and update the GUI. There is an option to add k-partite numbers to subgraphs.

The system includes features for finding the maximum clique and the maximum complete bipartite subgraph within a network, creating a layout to group Bipartite Network nodes into two columns, and finding clusters of well-connected proteins using methods like Walktrap, Fast Greedy, Edge Betweenness, Label Propagation, Leading Eigenvector, and Louvain.

Moreover, it integrates the map equation tool for evaluating community partitions' efficacy by predicting the average code length of a network using Huffman encoding. The system also implements the Infomap community detection algorithm, which uses the map equation to assess its effectiveness. Infomap begins by finding the local minima of a graph, recursively grouping these communities, and applying the same algorithm to these groups as was applied to the original network, continuing until no further improvements in the map equation are observed.

3 Purpose of New System

The project aims to enhance the functionalities of CyFinder within Cytoscape. A significant new feature is the introduction of support for directed weighted networks in both the Map equation and the Infomap algorithm, an expansion from the existing support for only undirected weighted networks. Enhancements to the current implementations of the Repeated Node Aggregation Algorithm and the Optimize Network Algorithm (described as Algorithms 3 and 4 in the "Mapping Higher-Order Network Flows" paper) are also part of this project. These improvements include the addition of a code length threshold to both algorithms and the incorporation of randomization into the Optimize Network Algorithm to avoid local optima. Furthermore, the project introduces the Fine-tuning Algorithm, detailed as Algorithm 5 in the "Mapping Higher-Order Network Flows" paper.

4 User Stories

This section presents the detailed user stories that were executed in this iteration of the CyFinder project. These user stories formed the foundation for the development of the project's features. Additionally, the section includes user stories that will be considered for future development.

5 Implemented User Stories

1. Update data structures and methods to consider the direction to support directed weighted networks.
2. Modify InfomapAlgorithm and MapEquation to support directed weighted networks.
3. Conduct initial testing to support directed weighted networks.
4. Implement a Threshold in code length to algorithm 3 and 3.
5. Added randomization to algorithm 4 to avoid local optima.
6. Conduct testing to in code length threshold and randomization.
7. Implement fine-tuning algorithm (Algorithm 5) described in *Mapping Higher-Order Network Flows Paper*.
8. Test fine tuning algorithm.

6 Pending User Stories

1. Implement coarse-tuning (Algorithm 6) from *Mapping Higher-Order Network Flows Paper*.
2. Implement repeated supermodule clustering (Algorithm 7) from *Mapping Higher-Order Network Flows Paper*.
3. Implement repeated submodule clustering (Algorithm 8) from *Mapping Higher-Order Network Flows Paper*.

7 Project Plan

This section outlines the planning process behind this project's execution. Utilizing agile development methods, it necessitated meticulous planning of sprints, which are elaborated in this section. Additionally, it details the selection of both software and hardware components used in the project.

8 Hardware and Software Resources

The following hardware and software requirements were necessary throughout the development of these CyFinder components.

Hardware:

- None: there was no need for specific hardware

Software:

- JAVA 13: the language that was used to write all our code for CyFinder.
- Eclipse: the IDE where we implemented all our features.
 - Apache Maven: used to generate JAR files.
 - Apache Ant: used to build the logic behind CyFinder.
- Cytoscape 3.9.1: core application where CyFinder is rooted.
 - Cytoscape API: powerful graph-based API used by CyFinder.
- GitHub: version control software used throughout the project.

9 Sprint Plans

Sprint 1

Sprint Planning Meeting Minutes:

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 1:00PM

End time: 1:00PM

The product owner has selected the following user stories for completion in the upcoming sprint, arranged in order of their priority.

- Download Cytoscape.
- Learn how to use Cytoscape.
- Get familiar with Cyfinder codebase.
- Learn how to implement Cyfinder with Cytoscape.

The team agreed to work on all the stories selected by PO.

Sprint 2

Sprint Planning Meeting Minutes:

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 1:00PM

End time: 1:00PM

The product owner has selected the following user stories for completion in the upcoming sprint, arranged in order of their priority.

- Understand cyfinder code.
- Use Cyfinder and understand what's happening.
- Run all the functions of Cyfinder.
- Understand how Cyfinder works.

The team agreed to work on all the stories selected by PO.

Sprint 3

Sprint Planning Meeting Minutes:

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 1:00PM

End time: 1:00PM

The product owner has selected the following user stories for completion in the upcoming sprint, arranged in order of their priority.

- Read tutorial.
- Run all Cyfinfer functions (provide screenshots)
- Debug functions.
- Keep studying the code.
- Better understand of CyFinder

The team agreed to work on all the stories selected by PO.

Sprint 4**Sprint Planning Meeting Minutes:**

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 1:00PM

End time: 1:00PM

The product owner has selected the following user stories for completion in the upcoming sprint, arranged in order of their priority.

- No specific user stories were provided by PO. PO gave us the task of deciding what to start implementing to the project.

The team agreed their willingness to work on the following user stories.

- Look at current Wishlist and study any possibility to add any to the project.
- Read documentation from previous team.

Sprint 5**Sprint Planning Meeting Minutes:**

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 1:00PM

End time: 1:00PM

The product owner has selected the following user stories for completion in the upcoming sprint, arranged in order of their priority.

- Update data structures and methods to consider the direction to support directed weighted networks.
- Modify InfomapAlgorithm and MapEquation and conduct initial testing.
- Come up with an approach to tackle the Wishlist items 2 and 3.
- Brainstorm ideas to incorporate other algorithms into InfomapAlgorithm.

The team agreed to work on all the stories selected by PO.

Sprint 6**Sprint Planning Meeting Minutes:**

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 1:00PM

End time: 1:00PM

The product owner has selected the following user stories for completion in the upcoming sprint, arranged in order of their priority.

- Implement Directed weighted networks.
- Implement new algorithms to Infomap.
- Tackle all backlogs from previous sprints.

The team agreed to work on all the stories selected by PO.

Sprint 7**Sprint Planning Meeting Minutes:**

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 1:00PM

End time: 1:00PM

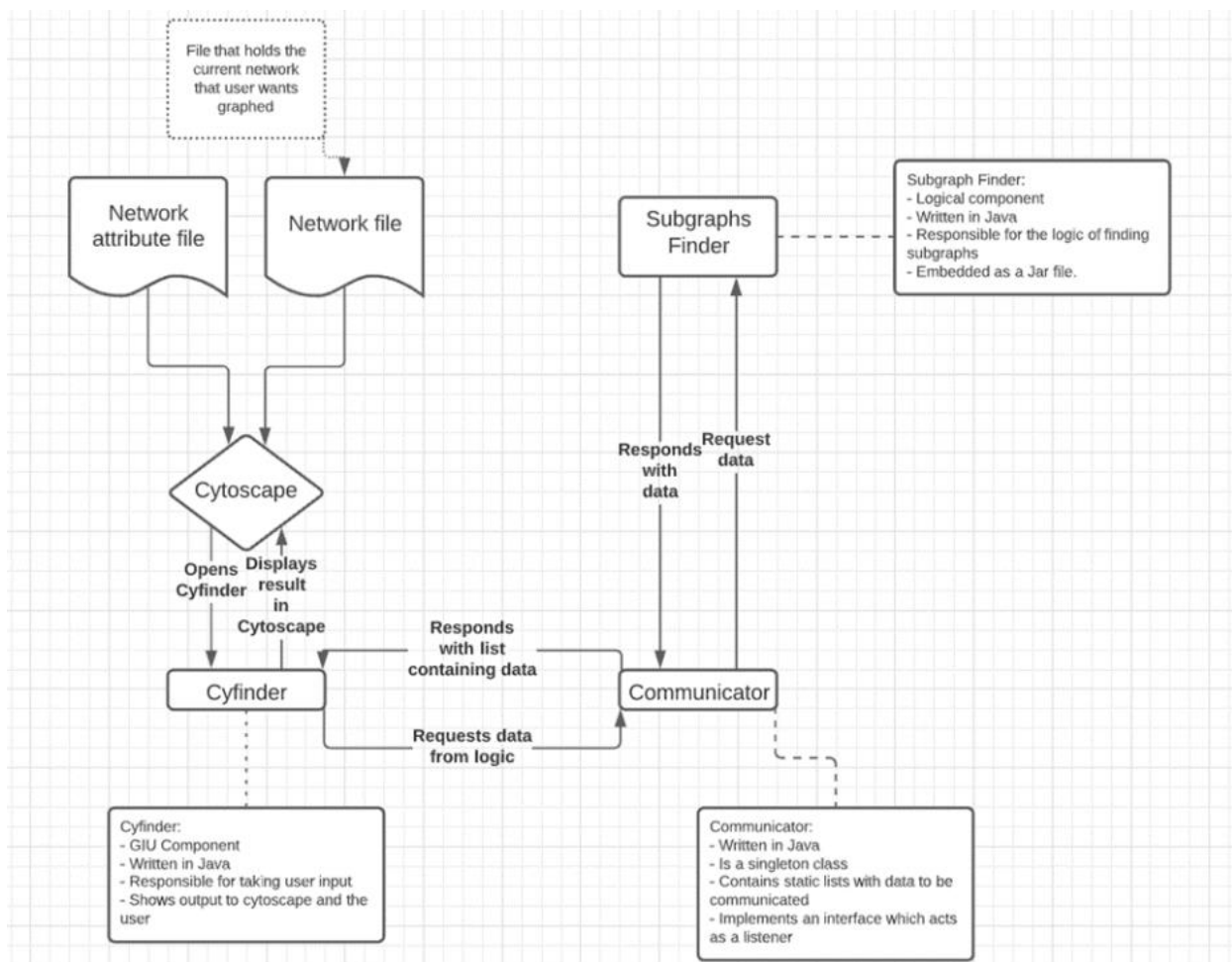
The product owner has selected the following user stories for completion in the upcoming sprint, arranged in order of their priority.

- Complete final deliverable
- Create video for showcase.
- Create video for final deliverable.
- Presentation for showcase.
- Poster for final deliverable.

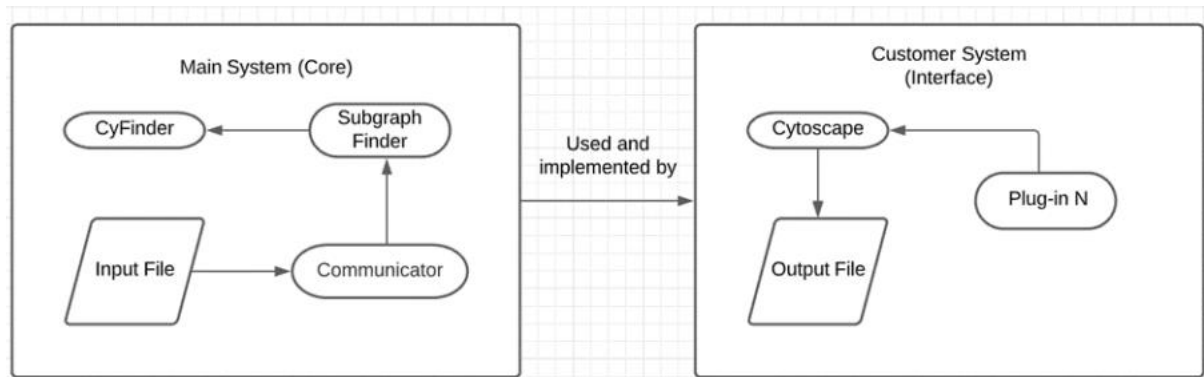
The team agreed to work on all the stories selected by PO.

10 System Design

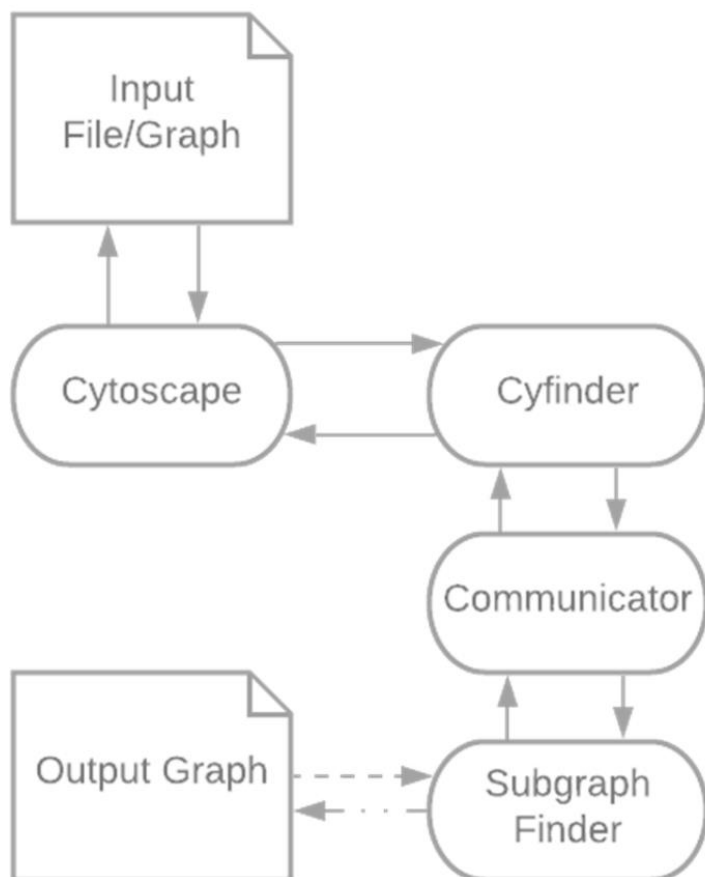
This section provides details on the design choices made for this project. It includes an outline and explanation of the architectural patterns used. A package diagram illustrating the entire system and explanations of the subsystems are also presented. Additionally, the section discusses the design patterns implemented in the project.



11 Project Dependencies Diagram



12 Design Patterns



13 Sprint Review Reports

Sprint 1 Review Meeting Minutes

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 7:30PM

End time: 8:00PM

After the presentation, the implementation of the following user stories were accepted by the product owners:

- Download Cytoscape.
- Learn how to use Cytoscape.
- Get familiar with Cyfinder codebase.
- Learn how to implement Cyfinder with Cytoscape.

Sprint 2 Review Meeting Minutes

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 7:30PM

End time: 8:00PM

After the presentation, the implementation of the following user stories were accepted by the product owners:

- Understand cyfinder code.
- Use Cyfinder and understand what's happening.
- Run all the functions of Cyfinder.
- Understand how Cyfinder works.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- NA

Sprint 3 Review Meeting Minutes

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 7:30PM

End time: 8:00PM

After the presentation, the implementation of the following user stories were accepted by the product owners:

- Read tutorial.
- Run all Cyfinfer functions (provide screenshots)
- Debug functions.
- Keep studying the code.
- Better understand of CyFinder

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- NA

Sprint 4 Review Meeting Minutes

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 7:30PM

End time: 8:00PM

After the presentation, the implementation of the following user stories were accepted by the product owners:

- No specific user stories were provided by PO. PO gave us the task to decide what to start implementing to the project.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- NA

Sprint 5 Review Meeting Minutes

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 7:30PM

End time: 8:00PM

After the presentation, the implementation of the following user stories were accepted by the product owners:

- Update data structures and methods to consider the direction to support directed weighted networks.
- Modify InfomapAlgorithm and MapEquation and conduct initial testing.
- Come up with an approach to tackle the Wishlist items 2 and 3.
- Brainstorm ideas to incorporate other algorithms into InfomapAlgorithm.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- NA

Sprint 6 Review Meeting Minutes

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 7:30PM

End time: 8:00PM

After the presentation, the implementation of the following user stories were accepted by the product owners:

- Implement Directed weighted networks.
- Implement new algorithms to Infomap.
- Tackle all backlogs from previous sprints.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- NA

Sprint 7 Review Meeting Minutes

Project: App Development for Analyzing Biological Networks

Attendees: Ananda M. Mondal, David Barahona, Gabriel Zavala, Riguens Brutus

Start time: 7:30PM

End time: 8:00PM

After the presentation, the implementation of the following user stories were accepted by the product owners:

- Complete final deliverable
- Create video for showcase.
- Create video for final deliverable.
- Presentation for showcase.
- Poster for final deliverable.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- NA

14 User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document, and other documents

14.1 How to install CyFinder

The CyFinder plugin comprises two distinct modules: the “CyFinder” module and the “Subgraph Finder” module. Each of these modules operates as a separate project. The CyFinder module manages the plugin's graphical user interface (GUI) within Cytoscape, while the Subgraph Finder module handles the plugin's logic.

The plugin functions by embedding the Subgraph Finder module as a jar file within the CyFinder jar file. This nested structure is created using Maven, resulting in a jar-within-a-jar setup. This configuration might initially seem complex, but the accompanying images and a step-by-step explanation are provided to aid in understanding.

14.2 How to develop

First, download the CyFinder and Subgraph Finder projects from the Google Drive folder, which your product owner should have shared with you. Then, import them as two separate projects into your preferred Integrated Development Environment (IDE). While all the steps in this tutorial are demonstrated using the Eclipse IDE, they should be compatible with any IDE.

14.3 Importing Subgraph Finder:

To begin, we'll import the "Subgraph Finder" project into Eclipse.

Importing Subgraph Finder into Eclipse:

Step 1: Click on 'File' in the toolbar, then select 'New -> Java Project' from the dropdown menu.

Step 2: Deselect the 'Use default location' checkbox, click on the 'Browse...' button, navigate to the directory where you downloaded and extracted the Subgraph Finder project from the shared Google Drive folder, and then click 'Finish'.

Building Subgraph Finder:

Next, we'll build the Subgraph Finder jar, which will later be utilized by CyFinder.

Step 3: Right-click on the 'build.xml' file in your Subgraph Finder project and select 'Run As -> Ant Build'. A successful Ant build will display "BUILD SUCCESSFUL" upon completion.

Step 4: Following a successful Ant Build, locate the generated jar file named "SubgraphFinder.jar" in the 'dist' folder of your "Subgraph Finder" project.

Note: Remember, any time you make changes to the "Subgraph Finder" project, often logical changes, you must generate a new jar file using this method. This jar file then becomes a dependency for the CyFinder project.

14.4 Importing and Building CyFinder:

Now, we will import the CyFinder project. It is a Maven project and should be imported as such, using the steps outlined below:

Note: If Maven is not installed as part of your IDE package, follow the instructions on Maven's official website for installation and setup.

Importing the CyFinder Project:

Step 5: Click 'File' in the toolbar and select the 'Import' option from the dropdown menu.

Step 6: Choose 'Maven' and then select "Existing Maven Projects".

Step 7: Press the 'Browse...' button, navigate to the directory of the CyFinder project that you downloaded and extracted from the Google Drive shared folder, and click 'Finish'.

Upon importing the "CyFinder" project, you will encounter an error in the "pom.xml" file related to the dependency "edu.claflin:finder:jar:3.0". This error indicates that Maven cannot locate this specific dependency, which must be installed manually for CyFinder to function properly. This

dependency is the “finder-3.0.jar” file created earlier using the Subgraph Finder project. If you followed the previous steps correctly, this jar file should be in the “dist” folder of the Subgraph Finder project.

Step 8: To use this jar file as a Maven dependency for the CyFinder project, place the jar file in the following directory on your system:
C:\Users\<YourUserNameHere>\.m2\repository\edu\claflin\finder\3.0

After placing the “finder-3.0.jar” file in the specified directory, you should notice that the previously mentioned dependency error in the CyFinder project, related to “edu.claflin:finder:jar:3.0”, is resolved and there should be no further errors.

Next, we'll test the setup by creating a jar file from the CyFinder project using Maven, which will then be imported into Cytoscape.

Generating a Jar File from the CyFinder Project:

Step 9: Right-click on the CyFinder project and navigate to 'Run As -> Maven install' from the dropdown menu.

If done correctly, the successful output should be visible in the console, indicating a successful Maven installation. This process will create a 'target' folder in your project's directory, containing two jar files. The file we need for Cytoscape is named “CyFinder-1.0.jar”.

Note: Each time you perform a Maven install, the old 'target' folder will be deleted, and a new one will be created. This step ensures that the latest changes in your project are included in the new jar file.

14.5 Developer Notes

- Enhancing Development Experience: A feature should be implemented that allows for the automatic building and importing of .jar files into the correct project directories. This would streamline the development process significantly.
- Testing Changes in CyFinder Project (GUI): Currently, the only method to test modifications in the CyFinder Project is to compile the jar file and deploy the changes in Cyfinder. This process needs to be carried out each time to ensure the changes are correctly implemented.
- JavaDocs Error Issue: There are errors related to JavaDocs in both the CyFinder and Subgraph Finder projects. Resolving these issues either by proper Java commenting or removing JavaDocs entirely would be beneficial.
- Developer Resources for Cytoscape: Further information for developers working with Cytoscape can be found at [Cytoscape Developer Documentation](#). Additionally, the Cytoscape API JavaDocs are accessible [here](#). These resources provide valuable guidance and reference material for development.

14.6 User Guide

Step 1: Installing the CyFinder App

- 1) Begin by opening Cytoscape and selecting 'Apps' from the toolbar. Then, click on “App Manager”.
- 2) In the App Manager, click on the “Install from File...” button located at the bottom left of the window. Navigate to the “CyFinder-1.0.jar” file's location.
- 3) Successful installation is indicated when the word “Installed” appears under Status. After this, you can close the window.
- 4) Finally, verify that CyFinder is accessible. It should be available from the 'Apps' toolbar dropdown menu in Cytoscape.

Step 2: Importing a Network into CyFinder

To import a network, click on the “Import Network from File System” button. Navigate to and select your network file.

Step 3: Selecting Your Network File

Upon selecting your network file, a window will appear displaying the network details. For instance, if you are using the “Lung_Cancer.txt” graph included with the source, it should show 82 nodes and 1062 edges. Confirm that all data is successfully processed by Cytoscape and click “Ok” to import your network.

Step 4: Viewing the Imported Network

Your network should now be visible in a format like the provided example. With your network imported into Cytoscape, you can now begin exploring the features offered by the CyFinder app.

14.7 Wishlist

1. Complete implementation of Infomap algorithm by adding other algorithms from the paper.
2. Add a Command Line Interface for the Threshold in algorithm 3 and 4 so the user can easily customize it easily.
3. Improve implementation for algorithm 5.

From previous team:

- 1) Add support for directed weighted networks in Map equation and Infomap algorithm.
- 2) Improve current implementations of Algorithm 3 and 4:
 - a. Enable small threshold for code length.
 - b. Add randomization to Algorithm 4
- 3) Contribute to implementation of Infomap algorithm by adding algorithm 5 from the paper.

15 References

Edler, D., Bohlin, L., & Rosvall, M. (2017). Mapping higher-order network flows in memory and multilayer networks with Infomap. *Scientific Reports*, 7(1), 1-13. doi: 10.1038/srep46689

Rosvall, M., & Axelsson, D. (2009). The map equation. *The European Physical Journal Special Topics*, 178(1), 13-23. doi: 10.1140/epjst/e2010-01179-1

D. Edler, A. Holmgren and M. Rosvall, The MapEquation software package, available online at mapequation.org.