COMP4801 Interim Report

Motif-G: A Motif-based Graph Analysis Platform

By

LI Boxuan (3035234698)

Supervised By

Dr. Reynold C.K. Cheng

Department of Computer Science
The University of Hong Kong

January 18, 2019

Abstract

Motif and clique are two important concepts in graph theory. A recently proposed concept motif-clique combines motif and clique together, providing abundant semantic information based on heterogeneous information networks. However, the existing work on motif-clique detection is rudimentary and lacks some important features. This project will continue the work based on the motif-clique concept and develop a fully functional motif based graph analysis platform together with a comprehensive algorithm. This project will develop several potential pruning strategies to accelerate motif-clique finding process and conduct extensive experiments to test performance under different workloads. In addition to a program with a command line interface, a web-based platform will be developed to provide users with visual results and rich interactions. This project will involve collaboration with bioinformatics researchers and explore how people can use the platform to learn human-gene associations and discover disease subtypes. Several improvements to the web platform and algorithm, and research on bioinformatics field are completed, while several advanced features and more advanced research problems will be resolved in the future.

Contents

Acknowledgment

Much appreciation for the crucial role of Dr. Reynold Cheng, who is the supervisor of this project. Many thanks to Dr. Jiafeng Hu who provided theoretical support and Mr. Budiman who provided a website prototype. Many thanks to Ms. Min Ou who offered help on the case studies on bioinformatics and Dr. Ruibang Luo who offered insights into bioinformatics field. A special thanks to Mr. Cezar Cazan who taught me numerous writing and presentation skills.

List of Figures

List of Tables

Abbreviations

m-clique Motif-clique
API Application Program Interface
HIN Heterogeneous Information network
HTTP Hypertext Transfer Protocol
MB Megabyte

1 Introduction

1.1 Background

1.1.1 Heterogeneous Information Network

Heterogeneous information networks (HINs), such as bibliographical datasets, are widely used and discussed in the field of data mining [1, 2]. Nodes of HINs are labeled, providing more abundant semantic meanings than unlabeled graphs [3]. Compared to homogeneous information networks, HINs distinguish different types of nodes and edges in the networks, consisting of rich semantic meanings of structural types of nodes [4].

1.1.2 Motif

A motif is essentially a small subgraph pattern, which is a foundational building block of complex HINs [5, 6]. Also known as higher-order structure, motif provides a tool to discover higher-order semantics of HINs [7]. It is widely used in graph analysis problems, such as graph clustering [8, 9], social network analysis [10].

1.1.3 Clique

A clique is by definition a complete graph, i.e., every two nodes in the clique are adjacent. Thus, a clique represents a set of nodes that are closely relevant (e.g., a clique in a social network can represent a group of close friends). Cliques have been widely studied in both research and industry communities. Usages of cliques include social network detection [10], gene group detection [11], and transportation network analysis [12]. A maximal clique is a clique that is not a subgraph of any larger clique.

1.1.4 Motif-clique

Hu et al. [7] proposed a new concept, namely motif-clique or m-clique in short, which incorporates motifs to the clique definition. Recall a clique is a complete graph based purely on edges, i.e. it is complete since every two distinct vertices are connected by an edge. A motif-clique, as a generalization of a traditional clique, is a complete graph based on a user-defined pattern, i.e. motif, rather than edges. An m-clique is, therefore, a *higher-order* clique based on a user-given motif. Compared to traditional cliques, which treats nodes with different labels equally, an m-clique can capture the

desired relationship among labeled nodes. A motif detection algorithm was proposed and implemented in C++ by Hu et al [7].

1.1.5 Motif-clique Query System

As Hu et al. [7]'s subsequent work, a basic functional online motif-clique query system was developed. Users can upload datasets following a specific format, or use a predefined dataset for demo purpose.

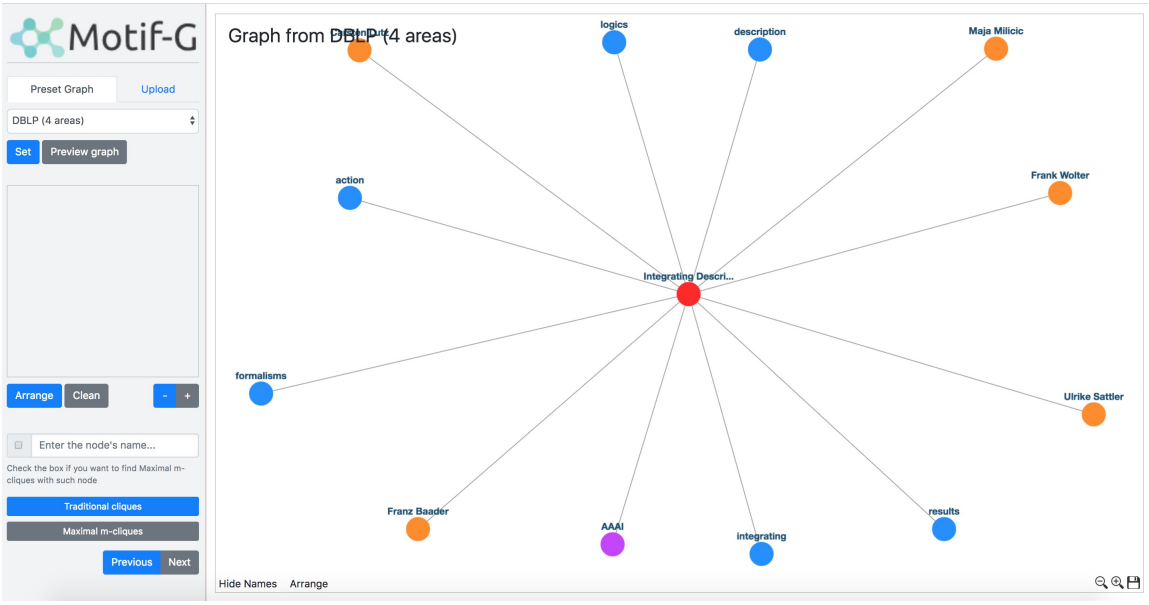

Figure 1: Motif-G - preview graph

As Figure 1 shows, after uploading or selecting a dataset, users can click *Preview graph* button to preview the graph. Before querying, users need to specify a motif, which is essentially a pattern that they want the clique to be based on.

As Figure 2 shows, a simple motif is defined on the left panel. There are four nodes in this motif, two of which are authors and the other two are papers. Each author is adjacent to both papers. This motif encapsulates the relationship among authors and papers, aiming to find co-authors who have at least two paper collaborations. Users can click *Maximal m-clique* button to see the result. The right panel on figure 2 shows an example of an m-clique based on the given motif. In this graph, two authors and several papers consist an m-clique. It is a motif-clique because, in this graph, any subgraph which consists of exactly two authors and two papers are connected in the same manner as the given motif. That is, if two papers and two authors are picked randomly in the motif-clique, each author must be adjacent to both papers and each

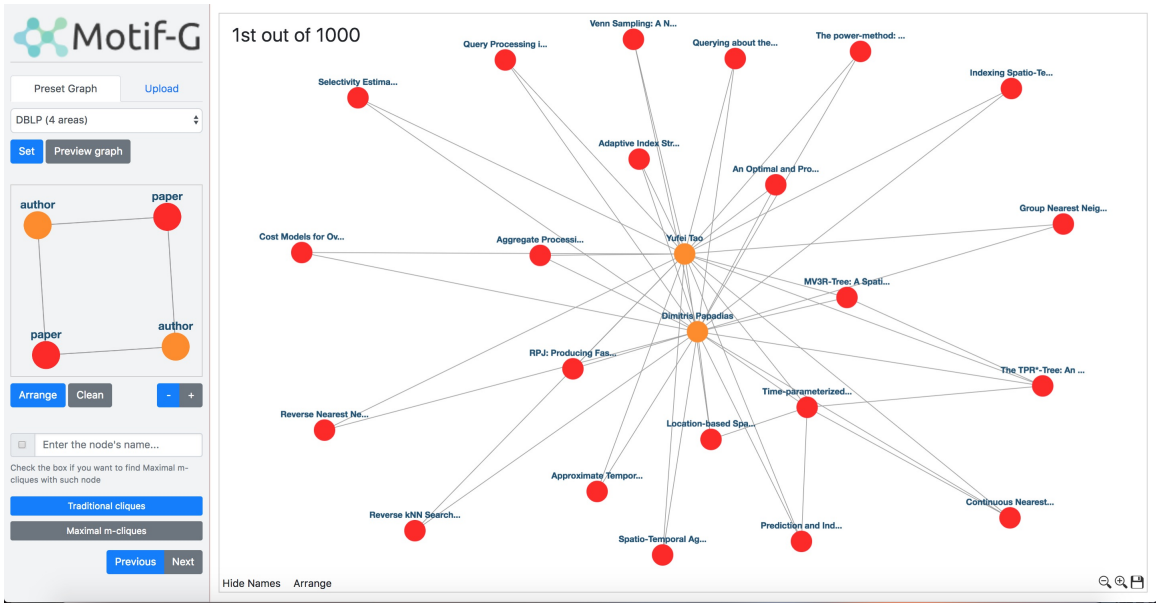

Figure 2: Motif-G - result of m-clique query

paper must be adjacent to both authors. It is a maximal motif-clique because no other paper or author in the dataset can be added to the motif-clique such that it is still a valid motif-clique.

However, the algorithm, together with the demo system, is incomplete and still has much space to improve. The algorithm is slow and the demo system only has basic features. Details would be addressed in the Methodology Chapter.

1.2 Objectives

This project aims to improve the algorithm and develop a fully functional motif-based analysis platform. The existing work by Hu et al. [7] will be utilized and extended. Currently, there are several limitations and problems with the algorithm and the web system. This project will resolve these problems and develop a more advanced and efficient motif-clique detection algorithm. Pruning strategies would be used to reduce search time cost in order to accelerate the the algorithm. A fully functional web-based platform will be developed. The user interface would be improved and prettified to provide better user experiences. Search options and parameters would be exposed and configurable. Instead of returning results in a batch after the whole process finishes, results would be returned in a stream to reduce the single response time. This project will also involve collaboration with bioinformatics researchers to apply the generic algorithm to bioinformatics research field. A conference paper or

journal article will be written if some interesting relationships among gene, drugs and human diseases are found.

1.3 Outline of the Report

This report is organized as follows. Methodology, including algorithm enhancement, web-based platform development, and case studies on bioinformatic datasets will be discussed in Chapter 2, where a discussion on the advantages and disadvantages of the proposed approach would be discussed. Work completed will be presented in Chapter 3. Current progress, including improvements to the algorithm and the web platform would be evaluated and limitations would be discussed. Chapter 4 illustrates the remaining work. Next steps including implementation of advanced features and case studies on bioinformatic datasets would be elaborated. The conclusion will be covered in the last chapter.

2 Methodology

The project consists of three parts, algorithm improvement, web platform development, and case studies on bioinformatic datasets. As a proof of concept, C++ would be used to implement pruning strategies for the algorithm. JavaScript and Java would be used to improve the user interface and implement advanced features on the web platform. Python would be used to preprocess bioinformatics datasets, which would be used to conduct case studies. Major work involved in this project would be discussed in this chapter while more details could be found in chapter 3 and 4.

2.1 Algorithm

Based on the analysis of the current algorithm, several potential improvements will be proposed. Pseudo-code for algorithms together with proof will be written, followed by actual implementation in C++. C++ is one of the most high-performant programming languages and is widely used for algorithm implementation. The deliverable would be an executable with a command line interface. Extensive benchmarks and experiments would be conducted to compare the efficiency before and after the improvement.

Moreover, this project involves collaboration with a bioinformatics research group at the Department of Computer Science, The University of Hong Kong. To fit datasets and generate more meaningful analysis and detections, the algorithm needs to be extended. For example, the current algorithm only supports unweighted edges, while bioinformatics datasets usually contain weighted edges. A configuration option would be added so that the algorithm would be able to support both unweighted graphs and weighted graphs.

2.2 Web Platform

The existing platform is written in Java and JavaScript. This project aims to improve the platform comprehensively, which involves both backend and frontend work.

The backend, written in Java, would be improved accordingly when the core algorithm is optimized. Java is a widely used language for backend web development and developers do not need to spend too much time with configurations. The drawback is that the algorithm is written in C++, thus the backend cannot incorporate the algorithm directly because it is difficult to integrate Java with C++. Therefore, the

algorithm was reimplemented using Java [7], which means future improvements to the algorithm using C++ need to be reimplemented using Java as well. This limitation is unavoidable, and luckily it does not bring too much trouble as the code logic are almost the same.

The frontend, written in JavaScript, would be improved to enhance user experience. Several features will be implemented to make the platform more usable and comprehensive. A configuration panel would be added so that users can specify parameters before searching.

2.3 Case Studies on Bioinformatics Datasets

The Motif-G platform has the potential to be used in the real world. This project involves case studies using bioinformatic datasets. Bioinformatics, as an interdisciplinary field, focuses on understanding biological data, such as identification of genes [17]. Motif-clique could be used for discovering the relationship between genes and human diseases. Common bioinformatics datasets, including DisGeNET [13], Reactome Pathway Database [14], NCBI E-utilities [15], DrugBank [16], and OPHID [18] would be utilized.

However, different datasets are in different formats. For example, as demonstrated in table 1 and 2, DisGeNET dataset and OPHID dataset have different schemas.

geneId	geneSymbol	diseaseId	diseaseName	score	source
10	NAT2	C0005695	Bladder Neoplasm	0.25	CTD_human

Table 1: Sample of DisGeNET

Dataset	SwissProt1	SwissProt2
SOURAV_MAPK_LOW	P63000	A0AUZ9

Table 2: Sample of OPHID

Consequently, those datasets cannot be combined unless they have compatible formats. Therefore, python would be used to clean and preprocess the datasets, merge them and import into the web platform. Python is a scripting language which is widely used as data processing and analysis tool due to its high readability. Useful data would be extracted from different datasets and merged into a single dataset so that researchers could see combined results.

3 Work Completed

Existing motif-clique detection algorithm, the code and relevant work were reviewed and evaluated. Drawbacks and areas for improvement were detected and documented. Project code was managed in a professional manner. A pruning strategy was implemented and the algorithm efficiency was improved. The web platform was enhanced.

3.1 Literature Review

Literature review shows that cliques are widely used in bioinformatics research. Clique-based gene analyses [19, 23] use cliques to find similar genes. Clique data mining with extension with filtering [22] is a novel approach which makes clique-based analysis more robust. Relationships between diseases and proteins are discovered and discussed with different approaches [20, 21]. These papers disclose that a clique is a powerful tool in the field of bioinformatics. Therefore, the m-clique platform could possibly be used to help bioinformatics researchers discovery new relationship between genes and human diseases.

3.2 Better Code Management

There were some inconsistencies between pseudo-code and real code, including naming conventions, function structures, etc. Additionally, documentation was missing and the code was not maintained in a professional manner. To improve readability and maintainability, documentation was added to the code. Documentation is very important for coding projects, as it helps people understand the logic and functionality of the code, thus making debugging easier. To improve development efficiency, redundant test data and auto-generated code were removed. Version control was introduced and the git tool was utilized to help keep track of changes to code and documentation. Copying and pasting code across different platforms or computers with different architecture can be tedious and prone to errors. Introducing version control and removing platform-dependent auto-generated code solved the problem and greatly enhanced portability.

3.3 Higher Algorithm Efficiency

To improve the existing algorithm is also part of this project. The existing m-clique search algorithm is not fast enough. Current experiments have shown that the search algorithm is much faster than a basic brute force search algorithm. However, it has

perceivable long latency when being used on the web platform. A huge graph usually contains tens of thousands of nodes and edges. Given a motif, up to thousands of maximal motif-cliques can be found. It is important that duplication is avoided.

Algorithm 1 GetMMC(U, C, NOT)

```
1: if C = ∅ then
2:     if NOT = ∅ then
3:         Output G[U] as a maximal m-clique
4:     return
5: while C ≠ ∅ do
6:     sample uniformly at random a node u ∈ C
7:     C ← C \ u
8:     if SubsetQueryProcess(U ∪ u, T) is true then
9:         NOT ← NOT ∪ u
10:        continue
11:    C_new ← Refine(U ∪ u, C)
12:    NOT_new ← Refine(U ∪ u, NOT)
13:    GetMMC(U ∪ u, C_new, NOT_new)
14:    NOT ← NOT ∪ u
```

The existing algorithm (Algorithm 1) uses a pruning strategy to detect and avoid duplication when adding a new node to the resulting graph, which significantly reduces repetitive results. Meanwhile, early pruning decreases program runtime tremendously by half or more. It uses a special data structure, namely *set-trie*, to keep track of interim results and detect duplication at an early stage. However, the set-trie can grow large very quickly when question complexity increases. Analysation and experiments showed that the set-trie data structure is not the optimal data structure to detect duplication as the whole set-trie has to be traversed to detect duplication.

To mitigate the overhead caused by set-trie structure, this project used a more efficient data structure. It utilizes the fact that when doing duplication check, only a small part of the graph, i.e. nodes that are directly connected to the new node, need to be checked. The pseudo code is as follows.

Algorithm 2 GetMMC(U, C, NOT, vector, count)

1: **if** $C = \emptyset$ **then**
2: **if** $NOT = \emptyset$ **then**
3: Output G[U] as a maximal m-clique
4: return
5: **while** $C \neq \emptyset$ **do**
6: sample uniformly at random a node $u \in C$
7: $C \leftarrow C \setminus u$
8: **if** $DuplicationCheck(u, vector, count, motifSize)$ is true **then**
9: $NOT \leftarrow NOT \cup u$
10: continue
11: $C_{new} \leftarrow Refine(U \cup u, C)$
12: $NOT_{new} \leftarrow Refine(U \cup u, NOT)$
13: $count \leftarrow Update(u, count)$
14: $GetMMC(U \cup u, C_{new}, NOT_{new}, vector, count, valid)$
15: $count \leftarrow Recover(u, count)$
16: $NOT \leftarrow NOT \cup u$

Algorithm 3 Update(u, count)

1: **for** $i \leftarrow 1$ to $vector[u].size$ **do**
2: $count[vector[u][i]] \leftarrow count[vector[u][i]] + 1$
3: return *count*

Algorithm 4 Recover(u, count)

1: **for** $i \leftarrow 1$ to $vector[u].size$ **do**
2: $count[vector[u][i]] \leftarrow count[vector[u][i]] - 1$
3: return *count*

Algorithm 5 DuplicationCheck(u, vector, count, motifSize)

1: **for** $i \leftarrow 1$ to $vector[u].size$ **do**
2: **if** $count[vector[u][i]] + 1 = motifSize$ **then**
3: return true
4: return false

Experiments showed that duplication check part of the old algorithm took up around 2% of the total execution time. The new pruning strategy reduced the number to 0.1%. Moreover, in extreme cases such as huge motifs are provided, the new algorithm can save more time and computation resources.

However, the limitation is that such optimization is bounded by the time needed by the duplication check. Compared to other portions of the whole algorithm, duplication check takes up a small part of the time. Therefore, the new algorithm has shown limited enhancement.

3.4 Platform Enhancement

The previous web platform was only a prototype and for proof of concept usage. To develop and deliver a fully functional and generic motif-based graph analysis platform, numerous features and improvements have been done.

3.4.1 Streaming Responses

Http request, which is widely used to communicate between frontend and backend, is stateless. The server (backend) cannot initiatively establish a connection with the client (frontend). When the client sends a request to the server, it has to wait until the server finishes processing the request and returns a response. However, under this platform's use case, the client usually wants a number of results, which takes the server much time to search and process. During search time, client gets no response at all, which brings about bad user experience. To address this problem, the server is enhanced and returns streaming responses instead. Server now responds as soon as there is some part of results found. Client displays those results immediately on UI while waiting for subsequent ones. In the end, server finishes searching process and notifies the client. This approach does not improve throughput, but greatly reduces single latency and improves user experience tremendously.

3.4.2 Union of Motif-cliques

Sometimes users want more complicated graphs based on motif-clique results. For example, as shown in figure 3, based on the given motif on the left, both motif-clique 1 and motif-clique 2 are valid and potential results. Considering they share a same node disease, there is possibility that an union of these two motif-cliques might be interesting.

A configuration option has been added and users can choose whether they want their results merged based on a same node of certain type. In the above example, users can specify that they want motif-cliques merged if they have the same disease.

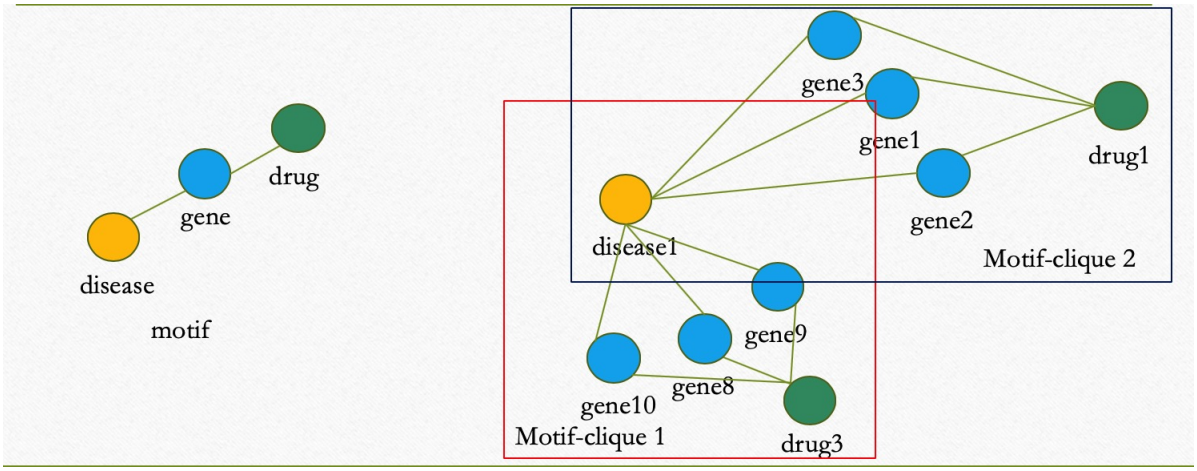

Figure 3: Union of motif-cliques

The combined graph is not a valid motif-clique anymore, but is a user-defined pattern on top of basic motif-cliques.

3.4.3 Restrictions on Number of Nodes

Motif-cliques are purely based on a given motif. However, when the given motif has little restriction, there would be much flexibility on results. For example, as shown in figure 4, given a simple motif consisting of three nodes and two edges, we want to find motif-cliques like the middle one, i.e. we want a single disease and many genes appearing on the results. However, unwanted results like the right one cannot be avoided.

Figure 4: Different results based on the same motif

To resolve this problem, an advanced option has been implemented and provided for users. Users could specify a threshold for each kind of nodes. When the number of nodes exceeds that threshold, the result would be dismissed and search space is

thus greatly reduced.

3.4.4 Edge Weight Threshold

In real world datasets, connection between nodes are not always reliable. Some might be more reliable and have larger weights, while others might be less reliable and have smaller weights. The existing algorithm does not support weighted edges, and as a consequence, the platform does not support weighted edges. Users could preprocess their data and filter out edges with smaller weights and then upload to the system, but it is cumbersome and they could not change filtering criteria dynamically. To address this problem, the algorithm has been enhanced and an advanced option has been provided for users. Users could specify a minimum edge weight on the UI before searching process starts. Edges with weights lower than the minimum edge weight will be ignored. Moreover, a sliding bar has been added to the UI. When observing and exploring results, users could use the sliding bar to change minimum edge weight dynamically. Edges lower than the specified number will be removed from UI, but still exist in memory for performance optimization.

3.4.5 Sorted Results

To help users find results of interests more easily, algorithm has been enhanced to support ordering. Instead of returning results in the default order, i.e. by chance, users can specify the ordering criteria on UI. For example, it is natural to assume motif-cliques with more nodes are more interesting than motif-cliques with fewer nodes. Thus, users can choose the option that they would like to see sorted results based on the number of nodes in the motif-clique. This helps users find interesting and useful results more quickly.

3.4.6 Category Visualization

In heterogeneous information networks, nodes are labeled and belong to certain groups, i.e. have some certain types, but sometimes type might be ambiguous and coarse. Nodes with the same type could sometimes be classified into different sub-groups. These fine-grained categorical information was ignored by previous works. A naive approach is to break down big groups into smaller groups, and treat nodes in a same big group but different small groups as different types. However, this approach has its own problem in that it dismisses the fact that these nodes belong to the same big group and might share some common characteristics.

To help users capture fine-grained differences, the platform has been enhanced to support categorical information, i.e. subtypes. If categorical information is provided, it would be collected, analyzed and displayed on a side panel on UI. Furthermore, users can click on categories and relevant nodes will be highlighted.

3.4.7 Other Functionalities

Other functionalities that have been added include search limit configuration, import and export, stop and reset, and search bar. Search limit configuration enables users to specify a search limit n, which means only first n results would be returned so that computation resource is saved. With Import and export functionalities, users can easily download results in serializable format from the platform, and also uploads existing results to the platform for visualization. With stop and reset functionalities, users can force the searching process running on the backend to stop, or reset the whole searching process. Search bar enables users to search and highlight one or more nodes on results for better visualization.

3.5 Research on disease subtyping

To conduct case studies using bioinformatics datasets, this project targets on finding disease subtypes, which is a novel field in bioinformatics research. In the genetic world, genes are related to various kinds of diseases and drugs. These relationships can be represented by graphs, where an edge between two nodes represents some relationship. An edge between a gene and a drug usually means the drug has some effect on that particular gene. An edge between a gene and a disease usually means the gene is one of the causes of that particular disease. In a graph consisting of diseases, genes and drugs, we would like to find subgraphs that conform to the following pattern:

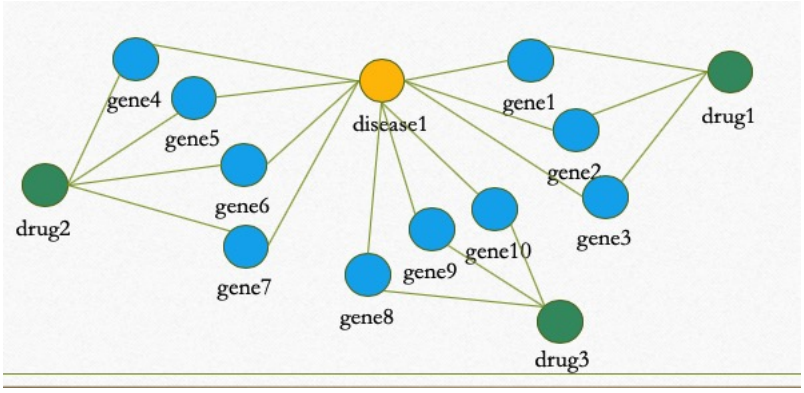

Figure 5: Target pattern

As figure 5 shows, a disease is connected to a bunch of genes. We can classify these genes into 3 groups, i.e. gene 1-3, gene 4-7, and gene 8-10, if we consider genes connected to a same drug as members in a same group. Consequently, we can infer that the disease can be classified into three subtypes, and each subtype can be cured with a unique drug.

As discussed in section 3.4.2, we designed a simple motif: disease - gene - drug, and combine results with same disease node. Results are shown in figure 6 & 7 below.

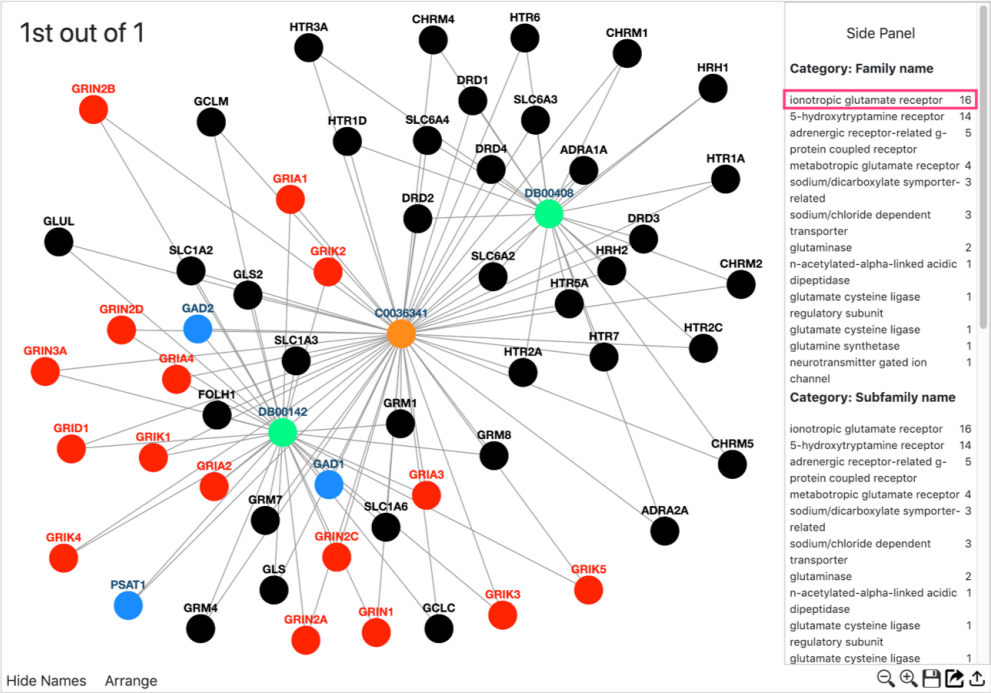

Figure 6: Disease Subtype 1

The orange node in the center, i.e. C0036341, is the concept id for disease schizophrenia. The green node in the bottom left, i.e. DB00142, is the id for drug glutamic acid. As shown in figure 6, genes on the bottom left are connected to C0036341 and DB00142. We can guess that schizophrenia has this particular subtype, which can be cured with glutamic acid. Note that red nodes are genes that belong to the ionotropic glutamate receptor family. As the figure shows, most genes on the bottom left part belong to the same family, which indicates that these genes are of similar characteristics. This consolidates our guess more, but the guess remains to be verified.

Figure 6 & 7 refer to the same result, with different cluster of nodes highlighted. The green node on the top right, i.e. DB00408, is the id for drug loxapine. Genes on

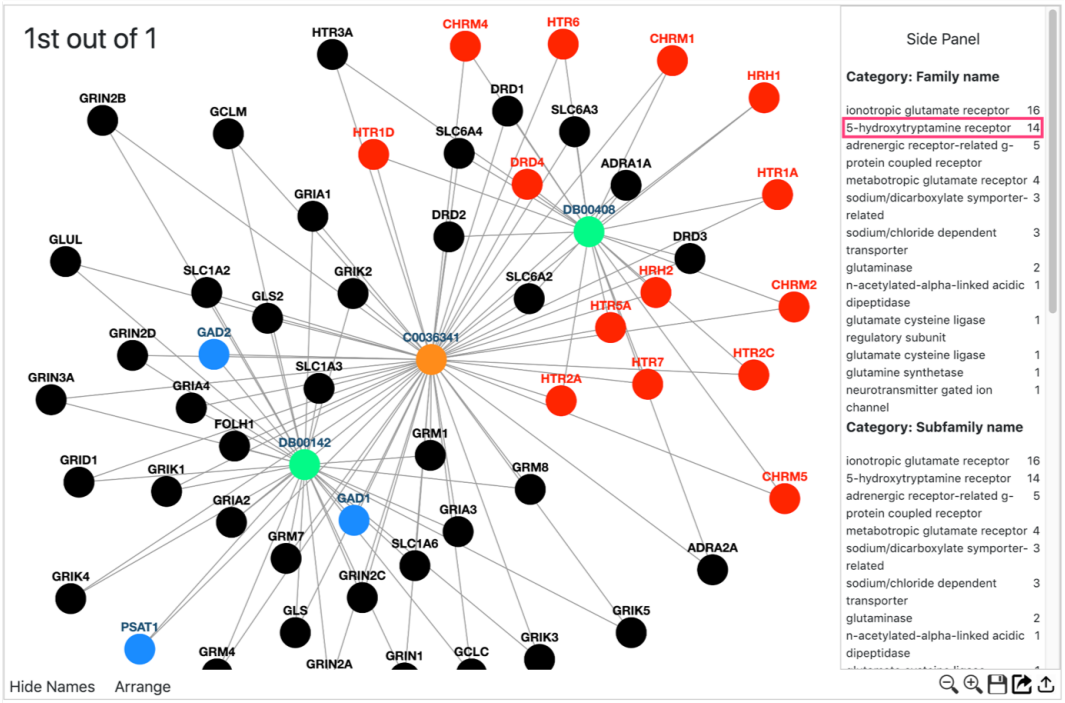

Figure 7: Disease Subtype 2

the top right are connected to C0036341 and DB00408. We can guess that schizophrenia has this particular subtype, which can be cured with loxapine. Red nodes are genes that belong to the 5-hydroxytryptamine receptor family. This consolidates our guess again, i.e. this disease might have two subtypes, each of which could be treated with a particular drug.

One problem is that there lacks evidence in academia which can support our guess. Regardless of the correctness of our guess, it provides insights and a new approach deep into the disease subtyping field, which was not deeply studied before.

4 Work Remaining

The algorithm and the platform have been greatly enhanced, and case studies on bioinformatics field have been conducted. There are still several tasks remaining.

4.1 Better Deployment and Usability

The platform is not ready for public use yet. It has stability issues, and the deploy process is cumbersome. In the next few months, deployment will be automated and enhanced. Usability will be improved so that every user including non cs experts would feel comfortable to make use of it.

4.2 Further Research on Disease Subtyping

This project has completed several research on disease subtyping and discovered some findings. The next step is to incorporate more datasets, design more motifs, conduct experiments and find relevant evidence based on literature reviews.

4.3 Find More Applications

To make sure the system is generic enough to be used in different disciplines, this project also aims to find more applications by making use of the platform as well as the algorithm. Besides usage in the bioinformatics field, if several more useful applications are discovered and analyzed, a demo paper will be written and submitted to a conference.

5 Conclusion

Motif-clique incorporates motif into clique definition, providing a new way to discover higher-order semantics of large heterogeneous information networks. The existing work is incomplete and has many limitations. This project aims to complete the motif-based graph analysis platform by improving the algorithm and enhancing the website. This project aims to provide a fully functional web-based motif platform, with which it would be flexible and convenient for researchers to conduct data mining on heterogeneous information networks. At the same time, this project involves collaboration with bioinformatics researchers and provides a generic web platform which can be utilized to detect and analyze abundant higher-order semantics in the bioinformatics field. This project improved the efficiency of the algorithm by using a new pruning strategy, but the improvement is very limited as it only increases the computation speed by approximately one percent. Several enhancements to the web platform are completed. Case studies on finding disease subtypes was conducted and some interesting findings were discovered. This project will involve more enhancements to the platform and deeper research not only on the bioinformatics field, but also on other possible areas.

References

[1] M. Ji et al. Graph regularized transductive classification on heterogeneous information networks. In ECML-PKDD, pages 570586, 2010.

[2] M. Ley. Dblp: some lessons learned. PVLDB, 2(2):14931500, 2009.

[3] Shi et al. A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering 29(1):17-37, 2017.

[4] Sun et al. Mining heterogeneous information networks: a structural analysis approach. Acm Sigkdd Explorations Newsletter 14(2):20-28, 2013.

[5] R. Milo et al. Network motifs: simple building blocks of complex networks. Science, 298(5594):824827, 2002.

[6] N. Przulj and N. Malod-Dognin. Network analytics in the age of big data. Science, 353(6295):123124, 2016.

[7] J. Hu et al. Discovering Maximal Motif Cliques in Large Heterogeneous Information Networks. In ICDE, 2019.

[8] H. Yin et al. Local higher-order graph clustering. In KDD, pages 555 564, 2017.

[9] A. R. Benson et al. Higher-order organization of complex networks. Science, 353(6295):163166, 2016.

[10] R. A. Hanneman and M. Riddle. Introduction to social network methods, chapter 11: cliques., 2005.

[11] G. A. Pavlopoulos et al. Using graph theory to analyze biological networks. BioData mining, 4(1):10, 2011.

[12] X. Yang et al. Bus transport network model with ideal n-depth clique network topology. Physica A: Statistical Mechanics and its Applications 390(23-24):4660-4672, 2011.

[13] J. Piero et al. DisGeNET: a discovery platform for the dynamical exploration of human diseases and their genes. Database 2015, 2015.

[14] Fabregat, Antonio, et al. The reactome pathway knowledgebase. Nucleic acids research 44.D1: D481-D487, 2015.

[15] Entrez Programming Utilities Help [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2010-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK25501/

[16] V. Law et al. DrugBank 4.0: shedding new light on drug metabolism. Nucleic Acids Res. 42(1):D1091-7, 2014.

[17] Wikipedia contributors. Bioinformatics. In Wikipedia, The Free Encyclopedia. Available from: https://en.wikipedia.org/w/index.php?title=Bioinformatics&oldid=870192795

[18] K.R. Brown, et al. Online Predicted Human Interaction Database. Bioinformatics, 21(9):2076-82, 2005.

[19] Navlakha, Saket, and Carl Kingsford. The power of protein interaction networks for associating genes with diseases. Bioinformatics 26.8: 1057-1063, 2010.

[20] Li, Xia, et al. The implications of relationships between human diseases and metabolic subpathways. PloS one 6.6: e21131, 2011.

[21] Yang, Lei, et al. Predicting disease-related proteins based on clique backbone in Protein-Protein interaction network. International journal of biological sciences 10.7: 677, 2014.

[22] Yang, Lei, and Xianglong Tang. Protein-protein interactions prediction based on iterative clique extension with gene ontology filtering. The Scientific World Journal 2014, 2014.

[23] Matsunaga, Tsutomu, et al. "Clique-based data mining for related genes in a biomedical database." BMC bioinformatics 10.1: 205, 2009.