CLUSTERING CANCER DRUGS ACCORDING TO THEIR MECHANISMS OF ACTION (MOA)

A DISSERTATION SUBMITTED TO THE UNIVERSITY OF MANCHESTER FOR THE DEGREE OF MASTER OF SCIENCE IN THE FACULTY OF SCIENCE AND ENGINEERING

2017

Ali Syed Abdullah

School of Computer Science

Contents

Word Count: 18550

List of Tables

List of Figures

Abstract

There is a wealth of data available related to drug ontologies which can be explored in order to extract important relationships between drugs. This is particularly important in cases where drugs share the same biomarkers such as a group of drugs targeting a specific type of cancer. This research aims to use Molecular and Gene Ontology matching between drugs, to highlight similarities and differences with respect to their Mechanisms of Action (MOA). The research was carried out on 'Melanoma' cancer-related drugs with a focus on developing a generic pipeline that can be extended to other groups of drugs. A numerical value of similarity was established between drugs using a combination of "Chemical Entities of Biological Interest (ChEBI)" for molecular ontology matching and "Gene Ontology Consortium" for Gene ontology matching. Data was then represented as a graph network where drugs represented nodes, relationships between drugs represented edges and weight represented level of similarity between drugs. First important relationship extracted was by ranking drugs with reference to their importance across graph centrality parameters including Degree, Closeness, Betweenness, PageRank and Eigenvector centrality. Unsupervised learning graph clustering techniques, including Chinese Whispers and Louvain Modularity, were applied to explore clusters between the drugs. Two important relationships were highlighted as a result of clustering; drugs that share the same dominant MOA were grouped together in a cluster and drugs that tend to have different MOA were placed in different clusters. Drugs were further ranked inside each cluster using centrality parameters of the network. Overall, drugs were clustered in groups, ranked in order of importance for both the whole network and inside individual clusters. The primary evaluation was done using silhouette coefficient, while secondary evaluation was done via general inspection by a domain expert. The confidence of results was strengthed by identifying similarities in clusters across multiple algorithms.

Declaration

I certify that no part of work referenced in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

Acknowledgements

All praises be to Allah, the Most Gracious, the Most Merciful and the All Knowing.

First of all, I would like to thank Allah, the All Mighty who gave me the opportunity, strength and wisdom to complete this project. Without His consistent help and guidance, it would never have been possible.

I would like to express my sincere gratitude to my supervisor Dr. Riza Batista-Navarro, for her patient mentoring, inspiring suggestions and unwavering confidence in my abilities. It has indeed been a privilege to have her guidance throughout the course of this research.

My deepest gratitude goes to my parents and family members for their prayers, consistent moral support and advises throughout my stay here at UoM. Their boundless love and encouragement mean a lot to me.

I am equally indebted to Dr. Ross King for his professional consultation in evaluating the results. As a domain expert, his feedback was extremely constructive. Furthermore, I would also like to extend my appreciation to the NaCTEM team, for providing a list of cancer event. The output of their research has been used as a foundation stone in this research.

I am grateful to my professional mentor Ms. Rabia Irfan. Her unfaltering support, valuable insight and encouragement have always motivated me to perform at my very best.

Last but not the least, my special thanks to my colleagues Mr. Hashir Kiyani and Ms. Devi Afriyantari, for their endless stimulating discussions at every stage of the research.

List of Abbreviations

Abbreviation	Full Form
ChEBI	Chemical Entities of Biological Interest
CW	Chinese Whisper
DDI	Drug to Drug Interaction
DPI	Drug to Protein Interaction
DTN	Delay Tolerant Networks
G-G	Gene to Gene Interaction
GO	Gene Ontology
LM	Louvain Modularity
LM[R=0.6]	Louvain Modularity with Resolution 0.6
LM[R=1]	Louvain Modularity with Resolution 1
MOA	Mechanism of Action
NaCTeM	National Centre for Text Mining
NME	New Molecular Entities
PFC	Purifying and Filtering the Coupling Matrix
PPI	Protein to Protein Interactions
STRING	Search Tool for the Retrieval of Interacting Genes

Chapter 1: Introduction

Natural Language Processing (NLP) is a relatively emerging field of Computer Science, though its emergence can be traced back to 1950 when Alan Tuning proposed the "turing test" [1]. Advancements in supervised and unsupervised machine learning algorithms over the last two decades have shown new prospects in making data driven decisions across a range of application areas. In NLP domain, training text mining models open room for more advanced application areas such as social media analysis, sentiment and machine translation, fighting spam, etc. This research focuses on using Unsupervised Learning for biomedical text analysis.

The amount of historical data available in the biomedical field is immense which presents certain challenges for its analysis. First of all, data available is predominantly unstructured and untagged i.e. not stored in relational databases from where specific things can be queried as desired. Though a lot of text is available for analysis in medical journals, the information is scattered within lines of text. So, the first challenge is extracting the information from multiple sources and storing it in a presentable and machine-readable format. Secondly, merely extracting the information is not enough as it is not new knowledge; the main task is to learn from this information i.e. drawing pertinent conclusions that are not apparently visible and hidden deep within the data.

A team of researchers have extracted information about cancer drugs and its interactions with other drugs and proteins at the National Centre for Text Mining (NaCTeM), the first publicly funded text mining center in the world. Researchers at NaCTeM used an online resource of biomedical papers, PubMed Library to extract drug references and their interactions into a single file. Their research acts as a building block for this research. Details of their research will be discussed later in this thesis.

1.1 Motivation

Cancer has one of the highest mortality rates. In 2012 alone, approximately 14 million cases were registered [2]. In next two decades following 2012, new cancer cases are expected to rise by about 70% [2]. There are different stages and types of cancer but usually, it is progressive, chronic and has a mild phenotype. The impact of cancer on human race is

overwhelming and the above statistics are a testimony to this fact. It not only highlights its significance but also makes it an important area for further exploration/research.

In 2010, only two New Molecular Entities (NME) were approved by the United States's Food and Drug Administration Authority to treat cancer. The maximum number of NMES registered raised to 11 in 2012 between the period of 2010 to first quarter of 2015 [3]. Failure of drugs has usually risen in phase 2 and 3 of testing due to under achievement of desired therapeutic effect or toxicity problems [4]. There is a need to explore new techniques in drug discovery to improve these numbers. There has also been a significant shift in research trends from focusing on selective drug target to multiple target approach[5].

Hence, the motivation of the idea is to involve a more generalized novel approach to studying drug relationships by using similarities in drug ontologies in order to highlight a cluster of drugs that share the same MOA, rank dominant drugs in cancer research and rank dominant drugs sharing the same mechanism of action (MOA). It is hoped that uncovering these relationships will increase our current knowledge of cancer drugs and open new gateways for further research.

1.2 Aim

To cluster cancer-related drugs based on their drug ontologies in order to highlight similarities and differences in their mechanism of action (MOA) followed by ranking all cancer drugs (in their order of importance) and ranking top drugs inside individual clusters as well.

1.3 Overview

List of drugs and other important drug relationships are extracted from the input file provided by NaCTEM. The goal of the project is to cluster these drugs into different groups based on the similarity they project with respect to their MOA. The criterion of determining similarity in MOA is based on two types of drug ontologies; their molecular structure and gene ontology. Using the two ontologies, a numeric value of similarity is established that varies from 0 to 100. The numeric value is calculated by first obtaining both molecular and gene drug ontology

trees for drugs in comparison and then finding out the overlapping terms in their ontological trees. Exact technicalities of calculating this numeric value would be established later.

Once pairwise similarity of drugs is established then these results can be shown as an undirected and weighted graph network. Drugs inside the graph structure are then ranked using five different centrality parameters including Degree, Closeness, Betweenness, PageRank and Eigenvector Centrality.

To explore hidden patterns of groups inside graph structure, three different unsupervised graph clustering algorithms are applied including Chinese Whispers, Louvain Modularity with Resolution 1 and Louvain Modularity with Resolution 0.6.

Drugs are ranked inside each individual cluster of different experiments using same five centrality parameters used before to rank drugs for the whole graph network.

1.3.1 Expected Outcome

Expected outcome of research is to highlight vital relationships between drugs. Primarily:

1. Highlighting groups of drugs that are similar according to their MOA.
2. Highlighting groups of drugs that are different according to their MOA.
3. Providing general ranking of all cancer-related drugs available in the data set.
4. Providing ranking of drugs that corresponds to a specific MOA.

1.4 Objectives

1.4.1 Learning Objectives

1. Research and select distinctive ontologies that would become the criterion of similarity between drugs.
2. Devise a practical method to calculate similarity based on the ontologies selected. This includes understanding large databases of ontologies and then calculating the similarity between drugs based on each individual ontology.
3. Exploring unsupervised graph clustering techniques including but not limited to Chinese Whispers and Louvain Modularity.

4. Exploring state of the art tools already available to implement the research including developing an application in Python using Pandas Library and using graph visualization tool, GEPHI.

1.4.2 Deliverable Objectives

1. Implement a text mining workflow that integrates several sources of ontologies together.
2. Extract ontologies of each drug and calculate pairwise weight (similarity) with all other drugs in the data set for different ontologies.
3. Clustering of drugs into groups of similarity and publish graphs for visual understanding.
4. Report the performance of using different clustering algorithms such as number of clusters detected, distribution of clusters and the percentage accuracy of the cluster etc.
5. Report recommendations for future work/research.

1.5 Thesis Breakdown

Chapter 1: Introduction- gives a generic introduction to the research, while explaining the motivation behind it. Expected outcomes, aim and objectives of the research are also established here.

Chapter 2: Background- identifies and describes different components which are used as building blocks for the research. Specifically, it explains:

1. What is MOA?
2. Describes the source of drug data set.
3. Presents the two drug ontologies that are used to find similarity between drugs.
4. Introduces graph clustering algorithms that are used for unsupervised learning.
5. Introduces centrality indices that are used for drug ranking.

Chapter 3: Literature Review- explains the extant literature concerning research methodology of using drug ontologies for cancer research. Furthermore, relevant literature with respect to the clustering algorithms used in this research and other similar machine learning approaches used for cancer research is also shared.

Chapter 4: Methodologies and Implementation- aspects related to project flow and research methodology employed are discussed. Important aspects are enumerated below:

1. Details about scope of the project.
2. All input data sources including NaCTEM input file, databases of molecular and gene ontologies.
3. Choice of relevant tools for data processing and data visualization.
4. A summary of experiment flow.
5. Details of individual implementation steps.
6. Establishment of evaluation parameters.

Chapter 5: Results and Analysis- some important points discussed in this section are:

1. General statistics of the graph network are calculated for different data sets.
2. Results of drug ranking across centrality parameters are shown.
3. Clustering results across different algorithms are shown with analysis of the results.
4. Top drugs inside individual clusters are also listed with respect to overall centrality parameters.
5. Evaluation results are discussed.
6. Views on future of research are discussed including recommendations and improvements.

Chapter 2: Background

2.1 Mechanism of Action

As already stated, the aim is to cluster drugs together on the basis of their MOA. MOA is a broad term and can be used interchangeably with different words depending on the context. In the biomedical domain, it refers to "biochemical interaction" that results in a drug producing its desired therapeutic pharmacological effect [6]. It usually involves an interaction of drug and molecular drug targets such as enzyme, receptors, ion channels, transporter proteins etc. to which a drug binds [7]. These reactions can cause two things; drug can either stimulate receptors (Agonists Drugs) or hinder stimulation of a receptor (Antagonists Drugs). For instance, HMG-CoA reductase (a rate controlling enzyme of the mevalonate pathway) inhibitors give improved regulation of cholesterol biosynthesis thus reducing the occurrence of heart diseases, asthmatic patients can now enjoy a better quality of life due to Leukotriene receptor antagonists, and the list goes on and on [8]. After having gained a basic understanding of MOA, let us have a look at their importance.

1. Generally, if drugs have similar MOA, then they share the same inducing effect; either good or bad which can be used to foresee issues concerning clinical safety. For example in case of anti-inflammatory drug development, targeting cytoplasmic membrane or electron transport chain is more likely to cause toxicity problems [9].
2. In depth understanding of MOA gives us an insight as to how a drug affects a drug target, which acts a stepping stone to the synthesis of new drugs. Similar drugs can be synthesized to replicate these reactions.
3. In some cases, it can lead to delivery of more customised medication for better treatment. For example, [10] mentions that "Trastuzumab" drug is known to target "HER2" protein but only a particular strain of breast cancer contains this protein. Patients can be screened for "HER2" and if the protein is found then a customized treatment plan can be recommended using Trastuzumab.
4. Cancer patients have often been advised a cocktail of drugs targeting multiple drug targets to ensure a small alteration in tumour DNA will not result in cancer treatment failure [11].

How does this research fits in to the bigger picture? Clustering of drugs according to their MOA will divide the drugs into groups that share common mechanism of action. This provides a stepping stone for cocktail generation as different clusters should target different MOA, thereby inhibiting multiple targets simultaneously. It will also help doctors make better

decisions, for example if a drug does not work well in the first round of chemotherapy then he may choose drugs for the second round from some other cluster. Likewise, distinctions and similarities between groups of drugs should highlight new insights in their MOA. In this research, the focus will only be on clustering of drugs. However, further biomedical analysis may be carried out to experimentally verify the relationships.

2.2 Cancer Research

There have been impressive breakthroughs in modern medicine over the last century. By focusing specifically on the MOA i.e. receptor subtype profile, better medication is now available for HIV, osteoporosis, migraine headaches etc. [8]. However, there still exists a lot of room for improvement in the realm of drug discovery particularly for diseases like cancer which still presents a potent challenge. Cancer is not something very rare; according to a research by British Journal of Cancer, 1 in 2 people will get cancer in their lifespan if they are born in England after 1960 [12]. There are two main troubles with cancer treatment. Firstly, its ability to evolve i.e. if a single drug receptor that is being targeted changes, the whole treatment for cancer fails. Hence the use of cocktail of drugs to target multiple drug receptors. Secondly, some tumours appear at places where their extraction from the body can become life threatening for the patient.

2.2.1 PubMed

A text mining approach to drug discovery can only work if there exists enough text mining data available to mine. At the moment, there is huge amount of data available on cancer research across different platforms, but one of the single biggest libraries on biomedical journals is PubMed with over 27.3 Million records, with 16% of them associated to cancer entries as of 2016 [13]. For Text Mining community, it is indeed a decent starting point.

2.2.2 Argo

After having established a source for text mining data, the second vital aspect is the selection of the data mining platform. There are many text mining platforms with their pros and cons but the best-suited seems to be Argos; developed by a team of researchers at NACTEM. It is a web based workbench with a modular approach to mining. It provides ability to drag and

drop text mining modular components and combining them together to create one seamless text mining pipeline as per requirement of a specific case study[14].

2.3 Drug Ontologies

We need to establish certain parameters that would act as a measure of similarity between drugs. The most obvious choice for this is to use known drug ontologies. There are a number of different classification of drug ontologies based on their biological function, molecular structure and physico-chemical properties [15]. This research will make use of Chemical Entities of Biological Interest (ChEBI) database for extracting molecular ontologies and Gene Ontology (GO) database for extracting gene ontologies of drugs. A brief introduction to both databases is summarized below:

1. Chemical Entities of Biological Interest (ChEBI) contains a dictionary for small molecular entities that are manually annotated [15]. Small molecular entities refer to synthetic or naturally occurring compounds that are unique. ChEBI does not contain molecular entities encoded by the genome. Some important characteristics of the database are:
    a. These entities are connected to each other to highlight a parent-child relationship to form a structured tree where parent is more generalised and child is a specialised term.
    b. There are four different types of ontology trees based on varying properties.
        i. Molecular Structure based on connectivity and composition.
        ii. Biological Role based on biological contexts such as coenzymes, antibiotics, hormones etc.
        iii. Application based on intended use such as fuel, pesticides, drugs etc.
        iv. Sub atomic particle based on categorization of particles smaller than atom.
    c. There are nine parent-child relationships in ChEBI. Few important ones are listed below:
        i. Is a: Highlighting relationship between specific and generalised term.
        ii. Is part of: Highlighting relationship of belonging between the two entities.
        iii. Is conjugate acid/Is conjugate base: Highlighting acid/base relationships that are acylic.
        iv. Has functional parent: Highlighting relationship that molecular entities in comparison are derivative of each other if some functional modification is done.

2. Gene Ontology (GO) is an effort to streamline the annotation and representation of controlled vocabulary for gene and gene products [16]. It is part of a larger effort by Open Biomedical Ontologies to control vocabularies shared in medical and biological domain. Some important aspects of GO are:

a. Terms are organised in a tree structure, where connection between entities represents some sort of “observable relationship”. Parent represents a more generalised term in the hierarchy whereas the child represents a more specific term.
b. A tree usually starts from a gene product or gene and leads up to some root domain. The relationships are directed acyclic graph where each term can point to multiple terms from different domains.
c. The tree structure leads up to three main root domains.
    i. Cellular component: Structural component of a cell or gene product group.
    ii. Biological Process: A development of events accomplished by involvement of multiple molecular functions.
    iii. Molecular Function: Referring to molecular activities such as binding, catalytic, activating etc.

2.4 Graph Theory

A graph can be defined as a visual representation of relationships between terms. A graph is composed of two parts; nodes and edges. Nodes are the actual terms whose relationships are to be represented. Edges are links between nodes defining the relationships.

Edges have further two properties; they can be directed/undirected and weighted/unweighted. Weighted graph has a scalar value assigned to each edge which indicates the strength of connection between the nodes connected by the edge. In unweighted graph, all edges are considered to have equal strength of connection. In a directed graph edges are directional, pointing from parent node to child node, whereas undirected graphs do not have any such distinction.

Relationships between entities that interact together can be represented in a graph structure, where entities represent nodes and edges represent relationship between entities. Representation in graph structure leads to two main advantages. Firstly, it can represent the bigger picture of how different nodes interact rather than just focusing on parent-child individual edges. Secondly, further analysis of graphs can be carried out to highlight identifiable structures based on different parameters. One of the ways to identify structures is based on connectivity of edges. Groups of nodes that are densely connected with each other and sparsely connected with rest of the graph can be grouped together. The idea of grouping nodes based on their connectivity is called clustering.

Let us now review how graph theory fits the case study being done in this research. First step would be to establish a level of similarity (weight) between drugs based on their drug ontologies. The output can then be represented in a graphical form where nodes represent drugs, edges represent a connection between drugs and weights represent strength of the relationship between drugs. No edge between two drugs signifies that drugs are unrelated to each other.

2.5 Clustering Algorithms

At this stage, the major challenge was to cluster the drugs together to find communities of similar drugs inside the graph structure. Community refers to a collection of nodes inside the larger graph structure that are sparsely connected to the rest of the network and more densely connected to each other [17]. These communities are unique as they may have significantly different local parameters compared with each other and when compared with average parameters of the graph, which include Betweenness centrality, Degree centrality, clustering coefficient etc. [18]. To identify communities, there are two major optimisation problems in clustering; firstly, the weight of edges and secondly, the connectivity of the structure. Two algorithms are used for this purpose, Chinese Whisper [19] and Louvain Modularity [20]. The choice of clustering algorithm is not random and will be explained in subsequent sections. At this stage, it is important to understand that 'cluster' or 'communities' refers to the concept of highlighting similar group of nodes and will be used interchangeably in this thesis.

2.5.1 Chinese Whisper

Chinese Whisper (CW) is graph clustering algorithm designed for unidirectional graphs that may be either weighted or unweighted [19]. The algorithm has been successful across multiple applications in NLP related tasks such as "language separation, acquisition of syntactic word class and word sense disambiguation" [21]. Since 2005, it has been widely accepted by NLP community and used across different domains. It best works for graphs that have small world property i.e. small average distance between arbitrary nodes and high clustering coefficient.

2.5.1.1 Steps

1. All nodes inside the graph structure are randomly assigned different classes i.e. number of nodes equal to number of classes.
2. Then, in a randomized fashion, any node is selected inside the structure and the node adapts the highest class among its immediate neighbours. If there is a tie among two or more neighbouring classes, any class is chosen among them. The step is repeated for all nodes inside the graph. This is marked as one iteration.
3. The algorithm undergoes multiple iterations before it either converges or comes to a state where it is oscillating between two or more states. To avoid the oscillation forever, a threshold for the number of iterations is set. A graph with 10000 nodes does not change much after forty to fifty iterations [21].

2.5.1.2 Properties

1. Hard partitioning: The property signifies that each node in CW cluster belongs to a particular class. Unlike soft partitioning where, there exists a probability distribution for likelihood of a node belonging to different classes.
2. Randomized: The result of communities identified inside a graph is different each time algorithm is applied. Reasoning for this behaviour stems from the fact that algorithm employs randomization in one of optimization steps resulting in different outcomes each time algorithm is applied. Nevertheless, the results are not totally the same but high degrees of similarity can be seen if the algorithm is applied multiple times. Moreover, the effect of randomisation is also inversely proportional to the size of graph.
3. Flat clustering: Clusters produced are unique without having any structural relationship to each other.
4. Time Linear: Computational time is quick even for high number of nodes which are densely connected.

2.5.2 Louvain Modularity

Louvain Modularity (LM) is a community detection algorithm inside a graph structure based on greedy optimization. The method optimizes network modularity, which is a measure of connectivity of a network. Modularity has a scale of -1 to 1 which is a density comparison of links inside community vs links connecting communities [20].

Greedy optimization algorithms try to find a locally optimal solution without considering the bigger problem and then try to find a global result based on local solutions [22]. Similarly,

rather than identifying multiple communities and optimizing their modularity all at once, the algorithm starts by optimising modularity on local nodes and these communities are combined. The step is repeated iteratively until we have optimized community structures.

2.5.2.1 Steps

1. All nodes are declared as a community.
2. Phase 1:
    a. Then for each node, modularity is calculated by placing it in neighbouring communities.
    b. Node is then moved to the community which results in highest increase in modularity. If no change in modularity then the node community is not changed.
    c. Step a and b are repeated sequentially until no further change is detected in modularity.
3. Phase 2:
    a. Once Phase 1 is completed, all nodes inside all communities are declared as new nodes.
4. Phase 1 and 2 are repeated again until there is no further change in community structure.

2.5.2.2 Properties

The algorithm shares all the properties of Chinese Whisper mentioned earlier. Algorithm output hinges on ordering of the nodes but the test experiments done by the author suggest that it is not very significant [20].

2.5.2.3 Advantages

Louvain Modularity has multiple advantages over other clustering techniques; it is exceptionally fast, simple to implement and does not require any additional parameters like definite a number of clusters. Unlike other modularity optimisation techniques, it does not suffer from “resolution limit” which gives it an unprecedented advantage [20]. Resolution limit is a common problem in modularity optimisation where communities that are smaller than a certain threshold are not detected; the exact threshold is dependent upon density of the structure and degree of nodes [23].

2.5.2.4 Resolution

This is possible due to intrinsic multilevel approach of the algorithm which is created by the phases of implementation. There is no guarantee though that the resolution limit can be completely avoided as first phase identifies individual communities which might get combined in phase 2 in later iterations. The ability to decompose the evolution of communities through multiple iterations is referred to as resolution. The parameter is finely tuned to display intermediate clusters. The value varies from 0 to 1. Here, 0 represents the initial stage of algorithm where all nodes are assigned to its own distinct class resulting in maximum clusters; equal to total number of nodes. 1 is the optimal resolution, where all possible clusters have been combined i.e. iterating phase 1 and 2 will not result in any further change in modularity. Though it is important to highlight that the intermediate clusters where resolution has not fully optimised also show promising results; highlighting sub communities where the difference between the communities is not clearly apparent [20]. The rate of convergence of modularity is very high at the start of the algorithm and it reduces as the resolution is increased; most of the clusters merge together in earlier iterations of the experiment and changes in resolution near optimization has minimal effect on clusters.

2.6 Centrality Indices

Important nodes in a connected graph structure can be ranked based on certain parameters called centrality indices. These indices represent influence of individual nodes from different perspective in the network. Influence is a generalised term; different indices focus on different aspects in which nodes can be ranked. Some commonly recognised centrality indices are Degree, Closeness, Betweenness and Eigenvector [24]. We also make use of another parameter page rank centrality. A brief introduction to these algorithms is given below:

1. Degree Centrality: The importance of each node is measured by the number of immediate neighbours it is connected with. Highest possible degree would be in case when a node is connected with every other node in the structure.
2. Closeness Centrality: Nodes are ranked based on shortest distance between each node and all other nodes in the structure [25].
3. Betweenness Centrality: Relevance of node is ranked with respect to the number of times a node crossed over for routing shortest path between each node and all other nodes in the network [25]. Hence, it is the number of times a node is crossed when calculating closeness centrality.

4. PageRank Centrality: PageRank was originally presented as web page ranking algorithm in [26] but it can be used for any generalised graph theory problem. It computes probability distribution considering the likelihood of randomly landing onto a page when traversing through pages that are linked together. The algorithm is designed to take into consideration the number of neighbours and quality of connections by these neighbours. Quality of the connection is decided based on degree of neighbour extended by that connection. PageRank is a popular ranking measure and has been one of the algorithms among many others that are employed by Google for search engine ranking. For undirected graph, the results are similar to degree distribution. In-case degree is consistent for all nodes in the structure then PageRank and degree distribution are exactly the same [27].
5. Eigenvector Centrality: Also a popular ranking method proposed in [28]. The base idea is the same as PageRank centrality measure based on influence of neighbouring node. If Node A is only connected to Node B and Node B has many neighbours which further have many other neighbours, then that implies that Node A is a strong node as it is very influential on the network [29]. Mathematically, it is defined as principal eigenvector of the adjacency matrix of a graph network. The base equation is represented by:

$$\lambda v = Av$$

Where λ it the eigenvalue, v is the eigenvector and A is the adjacency matrix of the network.

2.7 Graph Drawing Algorithm

There are no established standards for graphical representation of the network but there are a couple of things that need to be considered such as minimum overlap of edges, even distribution of nodes with respect to spacing and placement of nodes to highlight sense of structure. It is a widely researched area of interest with several feasible solutions. One of the classes of algorithms is force directed algorithms for plotting undirected graphs which have been widely accepted. Usually, force directed graphs optimize energy values in a graph by placing them together based on different spring like forces either attractive or repulsive or both between nodes. There are a number of algorithms that fall under this class e.g. Yifan Hu algorithm [30] , Kamada & Kawai algorithm [31] and Fruchterman & Reingold algorithm[32].

This research uses Yifan Hu algorithm, which places the nodes based on optimizing internode repulsive forces between nodes [33]. Only adjacent nodes are considered when computing repulsive forces to optimize the computational time. A problem with calculating optimized

energy between attractive and repulsive forces result is local minima points. To tackle this problem, Yifan Hu uses a technique called “adaptive cooling scheme” [30]. The algorithm starts with a high step size that reduces over time allowing faster convergence and avoiding local minima.

GEPHI platform has an implementation of Yifan Hu layout which is used for displaying clusters. A basic understanding of Yifan Hu has been given in this section but as this is not the main focus of this research, Yifan Hu algorithm will not be discussed in more detail.

Chapter 3: Literature Review

Extensive literature was reviewed targeting individual aspects of the research. Current trends in cancer research were explored while giving special attention to literature that was analogous to using drug ontologies to explore MOA. Focused areas of interest were types of the drug ontologies and their application in cancer domain, graph clustering techniques and their application in cancer domain, research related to protein clustering and drug ranking methodologies. Drug Discovery in cancer research is being targeted from many different angles. The basic idea of this research i.e. clustering drugs based on some similarity parameter has gained a lot of interest in recent years from researchers, though the exact specifications and research objectives vary greatly.

3.1 Research Approaches

One of the common approaches used by researchers is using gene expression gene profile for clustering either proteins or drugs. For example, a team of researchers clustered lung cancer drugs based on their gene expression profiles and drug sensitivity to multiple lung cancer cell lines [34]. The results suggested that one of the drugs acted particularly different than the rest of the drugs, hence might be useful in second line chemotherapy if it was not part of it initially. A similar attempt was made in [35], where the research focused on clustering 37 breast cancer drugs based on their sensitivity to 42 breast cancer cell lines. The results divided the drugs into 6 clusters, with 5 of them showing drugs with similar MOA while highlighting both new and known list of drug relationships between drugs and their sensitivities.

A somewhat different research was presented in [36]. In a more generic approach to "unweighted" graph clustering, researchers made use of drug bank database and imported all the drugs with their respective Drug to Drug Interactions (DDI). They made use of a clustering and visualisation tool GEPHI (More on GEPHI in Methodology Section) to find communities using Louvain modularity algorithm. 1141 nodes and 11688 links were divided into 9 different clusters. On a closer look, these clusters were broadly generalising a specific class of drugs such as one cluster represented drugs that specifically target immune system, another cluster represented drugs that work on nervous system, epilepsy related drug and so on. Only 15% of the drugs were not correctly classified, even on which the author urged that these drugs

need further study as the currently known DDIs for drugs might be inconsistent with their actual true DDIs. Nevertheless, research was able to identify known functional drug categories and relationships between them. Drugs were also ranked across five centrality parameters; Degree, Betweenness, Closeness, PageRank and Eigenvector centrality.

Research mentioned above is similar to our own research and hence, it validates the research methodology used in this research. Some similarities are; it clusters DDI relationships, it uses Louvain Modularity for clustering and it ranks drugs across five centrality parameters.

3.2 Choice of Ontology

3.2.1 Chemical Entries of Biological Interest (ChEBI):

The choice of clustering based on molecular and gene ontology was inspired by [37] which proposed how drugs can be classified into possible drug classes. The research proposes that all drugs can be divided based on 4 classes; their chemical structure, MOA, biomarker reactions and shared therapeutic outcomes. Based on this understanding of drug classes, drug ontologies were researched to find ontologies which best fulfil the set criteria. Chemical Entities of Biological Interest, a database of ontologies, an effort by European Molecular Biology Laboratory presented in [15] was the best fit as it had ontologies based on molecular structure, biological role, application and subatomic particles. Some finer intricate of ChEBI have already been established in the background section. This seems quite relevant to drug class distribution and was chosen as one of the similarity criterion between drugs.

3.2.2 GO

Calculating similarity between drugs merely based on ChEBI did not seem to be enough. Despite the fact that it satisfied all requirements of drug class but another fundamental aspect in which drugs can be categorized is drug targets. 400 out of 30,000 genomes in the human body are used for encoding protein for drug targets [38]. A lot of research in the area has been focused on clustering based on gene expression profiles. One of the examples of similar research is presented in [39] that discusses the use of both unsupervised and supervised learning approaches for analysis of gene expression profiles which can be used for cancer classification.

Another interesting research is presented in [40], which uses protein targets in ChEMBL (manually curated database of bioactive molecules) [41] and maps them to high level Gene ontologies by using QuickGO tool [42] which uses the GO database annotations. The research bridged the gap between chemical and biological information by mapping 6200 protein targets in ChEMBL to 300 GO terms. In short, their research focused on classifying different proteins under high level GO terms.

In this research, the author extended the idea to match drugs on the basis of similarity in their corresponding gene product ontologies. Gene ontologies extracted from GO terms become the second criterion of matching; first being molecular structure extracted from ChEBI.

3.3 Clustering Analysis

3.3.1 The Bigger Picture

Clustering algorithms have extensively been used in NLP related task. Recent developments in biomedical domain led to the availability of large amount of data to be processed.

One such example is microarray analysis, which has gained interest from researchers for unsupervised learning from data. Microarray analysis is parallel monitoring of gene expression levels in thousands of genes during a biological process across different environment or tissue sample (such as tumour sample). It is like taking screenshots of gene expression levels along different steps of the process [43]. Research carried out by [44] based on 8000 genes and 60 cell lines by National Institute of Cancer (NCI-60) using hierarchical clustering of microarray analysis showed that two of the breast cancer cell lines were similar to melanoma cell lines suggesting some sort of relationship between some breast cancers and melanoma. Some of the formerly known facts were validated and a new functional relationship came to light.

Another team of researcher did k-means clustering on gene expression data to understand the mechanism of cells; particularly understanding changes in mitochondrial proteins inspired by mitochondrial DNA depletion [45]. Their research revealed some vital relationships such

as how cells compensate for mitochondrial DNA depletion and identification of proteins that repair this particular DNA depletion.

It is also important to highlight that clustering is just one aspect of how ML based approaches are supporting research in NLP. Various other ML based approaches are actively being used such as supervised learning and neural networks in facilitation of cancer research. [46] uses supervised machine learning algorithm to train classifier to successfully distinguish between normal and tumour cell based on data of microarray analysis.

Another research across multiple institutions led to a promising breakthrough in drug discovery engines. The team trained a 7-layer neural network based on 6525 compounds profiled on MCF-7 cell line based on NCI-60 cell line assay data. Neural Network was then tested with 72M molecules in PubChem database to identify molecules that are most likely to contain anti-cancer properties. This resulted in 69 molecules, some of them with known anti-drug properties and some new anti-cancer molecules were also identified. More information can be found in [47].

Clustering/Community detection algorithms are very useful to evaluate large connected structures such as the ones that arise in analysis of world wide web, citation networks, social media networks, transportation networks, biochemical structures and other connected networks [48]. As previously explained, clustering is basically identifying groups of nodes based on the structure of network connectivity. Communities have a high density of connectivity inside the group of nodes and low density between other communities [49]. One of the commonly known community detection algorithm is Givan-Newman algorithm [50]. The algorithm focused on identifying “edges between communities” by shortest path between all nodes. The output of the algorithm is dendrogram, a hierarchical top down structure representing split of communities. Newman calculated spectral optimization of modularity which is reformulation of modularity into simpler matrix representation of eigenvectors [51]. Both these researches have been important breakthroughs in community detection. They have been used by different authors with slight variations to optimise the running time such as [52], [53] and Louvain Modularity [20]; one of the algorithms that is used in this research and is briefly explained below.

3.3.2 Louvain Modularity

Finer details of Louvain Modularity algorithm have been established in the background section. Focusing on current application areas of Louvain Modularity, the original author uses it in two experiments; Mobile Phone Network (2.6M nodes) and Web Graph (118M nodes) [20].

Another research [54] used it to evaluate and compare the algorithm with spectral partition algorithm called METIS [55] over twitter data set of 2.4M nodes and 38M links and Orkut data set of 3M nodes and 223M links. Based on criteria of intra partition edges, use of Louvain Modularity clustering with weighted graphs outperformed other techniques. Applications of Community detection are endless. One group of researchers used it in citation network [56] to do clustering based on ISI category classification. Another used it to do market analysis by clustering of retail transactional data in [57]. Social media analysis has many publications such as [58] which uses it to analyse Freesound data (audio clip sharing site), [59] uses LinkedIn data to explore how use of clustering data along with normal search method can be combined to improve search result relevance in social media websites and [60] uses three sources, Flicker, Live journal and YouTube data to improve efficiency of Delay Tolerant Networks (DTN).

Biological networks show a high degree of modularity; hence it is viable to use modularity based techniques for community detection such as Louvain Modularity and Markov Clustering to find communities in the network. Over the recent years, modularity based techniques have gained interest from researchers to modularise and study complex structures that occur in nature especially Protein to Protein Interactions (PPI) and brain networks. Modularity in networks can also be used to simplify parameter optimisation in-order to tackle curse of dimensionality in large scale biochemical models [61]. One such application case is induced apoptosis of CD95 to understand the transduction network; where modularity helped reduce 58 unknown parameters to just 18 parameters [62]. The intermediate results of Louvain Modularity algorithm portray useful information [20]. These results can be extracted by varying the resolution of the experiment. Most of the convergence occurs in initial phase which gives an advantage that if we extract results before the final stage, we would be able to identify super-clusters relevant to our final result. After doing some experiments and trying different resolution levels, 0.6 resolution was found to be optimal choice to extract sub-clusters while overall base clusters remain the same.

3.3.3 Chinese Whisper:

Chinese Whisper was first proposed in [19] as an unsupervised approach for multilingual text corpora sorting based on sentences. A weighted graph is constructed where nodes represent words, edges as associations and weights of edges represent concurrence of words. Their results suggested a high degree of success comparable to supervised learning approaches.

Realising the broader scope of the application of the algorithm, a same team of researchers came up with another research paper formally explaining the algorithm as a generic unsupervised graph clustering approach in [21] and its applications in NLP domain. Specifically, they performed experiments for Language Separation, Acquisition of Word Class and Word Sense Induction. CW has not been directly used for drug to drug clustering, although it has been used for gene to gene interaction (G-G) clustering, which has remained a key area of interest over the past few years. As clustering G-G relationship of a biological process can highlight molecular signatures related to a specific process and identify relevant biomarkers, this is one of the first steps towards identifying gene functions and it can also be used to identify drug targets.

A team of researchers presented a leader gene approach which clusters genes involved in a disease or cellular process using Chinese Whispers Algorithm [63]. The clustering is based on the data extracted from multiple sources including Search Tool for the Retrieval of Interacting Genes (STRING) [64] The weight of interaction between genes is calculated based on parameters extracted from STRING. They were able to cluster together classes of genes and also rank them according to the number of interactions of each gene. This is similar to our research with two key distinctions; this research focuses on drug to drug classification and weight between links is calculated based on drug ontologies extracted from CHEBI and GO Consortium. Another research proposes soft clustering algorithm; purifying and filtering the coupling matrix (PFC) [65]. A comparison is carried out between PFC, CW, C Finder [66] and Betweenness [67] for clustering Protein to Protein Interactions (PPI). Soft clustering means a node can belong to more than one class. Soft clustering can be advantageous in certain scenarios where a node behaves according to the environment such as context dependent protein functionalities. In unweighted graphs, the experiments suggest that PFC outperforms all other algorithms based on the comparison of average protein enrichments of clusters.

Chapter 4: Methodologies & Implementation

4.1 Scope Definition

1. Scope of the project has been limited to clustering of Drug to Drug Interactions (DDI) using drug ontologies extracted from ChEBI and GO.
2. For both ontologies, a tree depth level of three is chosen. The whole tree is not considered for matching as it is computationally impossible to compute and furthermore as we move towards the root of the tree the ontologies become more similar i.e. it will give higher matching for all the drugs.
3. Drugs and protein relationships are only limited to ARGO Event File.
4. Clustering of the network is limited to three algorithms; Chinese Whispers, Louvain Modularity with resolution 1 and Louvain Modularity with resolution 0.6.
5. Ranking of drugs is limited to 5 centrality measures; Degree, Closeness, Betweenness, PageRank and Eigenvector centrality.

4.2 Data

4.2.1 ARGO Event File

ARGO event file is an output of an event extraction pipeline constructed in ARGO that focuses on highlighting Drug to Drug Interactions (DDI) and Drug to Protein Interactions (DPI). Biomedical event extraction involves identification and extraction of structured and categorical association between biochemical entities [68]. Not only just drug names but some other relevant event features are also extracted based on information in the sentence about the interaction of entities such as event type, event trigger, etc. Table 1 shows key features of the ARGO data set with description of each term and unique categories belonging to that term.

ARGO event file contains a lot of features, though not all of these are used in our experiment. It is important to understand the data set in order to establish its authenticity. The file contains events extracted from 6529 melanoma related papers referring either drugs or proteins or both.

Term	Description	Category
Event ID	Unique Identification Key	
Event Type	Type of Reaction between Primary and Secondary Argument	26 Unique Categories
Event Trigger	Key word identifying initiation of the event.	
Primary Argument Text	Name of Entity (Drug or Protein)	
Primary Argument Ground ID	ID of ChEBI or UniProt	
Secondary Argument text	Name of 2nd Entity in (Drug or Protein)	
Secondary Argument Ground ID	ID of ChEBI or UniProt	
Negation	Highlights events which have a contradicting relationship between them	Boolean
Uncertainty	Value indicating whether event seems to be true from a linguistic point of view	Boolean
Confidence	Probability of an event being correctly extracted	Range 0-1.
Sentence	Sentence in which event occurs	Sentences in which event occurred
Paper ID	Unique Paper ID	6529 unique papers were scanned

Table 1: Key features of ARGO data set.

4.2.2 Event Types

There are three different categorize of Event Types in the event file.

1. Drug to Drug Interaction (DDI): Both Primary and Secondary Arguments are drugs. (Total Unique Instances: 155)
2. Drug to Protein Interaction (DPI): One of the Primary or Secondary Argument is drug and the other one is protein. (Total Unique Instances: 467)
3. Protein to Protein Interaction (PPI): Both Primary and Secondary Arguments are proteins. (Total Unique Instances: 8981)

Protein to Protein Interactions will not be clustered in the clustering process as we want to focus on drug to drug classification. Total unique drugs are 831. Total unique drugs with a protein relationship are 703.

4.2.3 Molecular Ontology Database [ChEBI]

ChEBI database is available for download in three formats “Flat file and Tab Delimited”, “Oracle binary table dumps” or “Generic SQL (Structured Query Language) table dumps”. In this research, we will use “Flat File and Tab Delimited” database; files are available in .tsv format. ChEBI database is extremely large and only a fraction of it will be used in this research. Let us discuss few important tables.

Vertex: Table describing each individual vertex/node:

Column	Description
Id	Unique Identifier
vertex_ref	ChEBI ID of drug
compound_id	Foreign key of Compound table, which contains more information about the drug name, definition, source, creation date, etc.
ontology_id	Foreign Key of Ontology Table

Table 2: Description of Vertex table.

Relation: Table describing relationship between vertices.

Column	Description
Id	Unique Identifier
Type	Type of ontology relationship
Init_id	Parent vertex
Final_id	Child vertex
Status	Status “C” signifies that entity has been manually checked by database curators anything else means it needs to be checked.
Drug_id	Empty Field for future development

Table 3: Description of Relation table.

4.2.4 Gene Molecular Ontology [GO Consortium]

GO Consortium database is available in a number of formats including .obo, .owl and SQL dump format. There are a number of third party applications that use GO data. Tools supported and developed under GO are AmiGO and OBO-Edit. GO database is also extremely large and like ChEBI, only a fraction of it will be used in this research. Few important tables are discussed below.

Dbxref: Table referencing terms from other databases.

Column	Description
Id	Unique Identifier.
xref_dbname	Reference Database Name from which the terms originally came from such as UniProt, PubMed, ChEBI, etc.
xref_key	ID of the entity from the reference database.
xref_keytype	Deprecated column
xref_desc	Deprecated column

Table 4: Description of Dbxref table.

gene_product: Table joining UniProt to gene Products.

Column	Description
Id	Unique Identifier
symbol	Short label of Gene Product referencing another referencing database.
dbxref_id	Foreign Key of dbxref table referencing original entity.
species_id	Species to which gene product belongs.
type_id	Term id of gene_product

Table 5: Description of gene_product table.

term2term: Table representing structural relationship between two entities i.e. each tuple represents an edge in the ontology graph.

Column	Description
Id	Unique Identifier
relationship_type_id	Relationship category between term1 and term2. [Categories are: Is a, Is part of, Is conjugate acid/Is conjugate base, Has functional parent]
term1_id	Node ID of Parent Node in the relationship
term2_id	Node ID of Child Node in the relationship.
complete	N/A-Term under development for future use.

Table 6: Description of term2term table.

4.3 Framework Language & Tools

4.3.1 Python Pandas

It became clear in the early stages of this research that conventional methods would not apply in this case owing to a large magnitude of the data sets.

1. Molecular and gene ontologies databases were very large; some having millions of tuples.
2. Calculations were computationally intensive, particularly the ones encountered in creating and matching drug ontologies.

3. Some of the queries were recursive which caused problems like memory leaks, long execution times and debugging also became a challenge under these conditions.

These problems present a challenge for normal SQL engine database even with using multiple indexes. Simpler queries of 'where' clause took few seconds and complex ones took few minutes but iterative calls to some queries made the system wait for long hours. SQL functions were used to resolve the issue but they also did not provide much improvement in execution time. To tackle these problems, an application was built in Python using Pandas Library. It is a data analysis library based on data structures. The uniqueness of the library is its capacity of in-memory database i.e. it calls the whole table in RAM thereby making searching and retrieval of tuples extremely efficient. The table can just be accessed by a variable called "Data frame" and the library provides multiple functions to do operations on the whole data frame simultaneously. Data frame can be loaded directly from a .csv comma delimited file or from a SQL database table. Due to these functionalities, Python Pandas seemed to be most viable option for data processing in this research. The library is free to use under BSD License.

4.3.2 GEPHI

It is an open source platform for analysis of data and visualisation, particularly the graph data. First introduced in [69], the tool has since been used in multiple journals to manipulate structures, discover new patterns and represent data. It has a plugin library which offers features such as clustering algorithms, graph layout options, node filter library, options to compute general statistics related to graph network and many other smaller features related to graph network processing. Further literature for using GEPHI in network analysis is available on [33]. It's still in active development with last stable release v0.9.1 in February 2016.Though, there were some major changes in v0.9.0 which meant that all the clustering algorithms needed some modifications to make it compatible with revised version. Some algorithms including Chinese Whispers were not working in the latest version (date of last check was 4th July, 2017). Hence, this research uses v0.8.2 which has all the major clustering algorithms in a working state. Development team of GEPHI was also reached out to ask the status of clustering algorithms in latest version and they responded that they may be able to complete it in next couple of months.

GEPHI is easy to use, both Chinese Whisper [21] and Louvain Modularity with variable resolution [20] are available as ready to use plugins. Other than that, it provides complete framework for data analysis. Notable features are data loader functionalities, layout options for visualisation, node filtering options based on multiple criteria etc. Owing to paucity of time, scope limitation of the project and completeness of the available tool, it was decided not to build a new implementation of the clustering algorithm. Instead, data processing was carried out in Python Pandas followed by clustering and visualisation in GEPHI.

4.4 Experiment Flow

Before discussing details of system flow, it is important to establish a general understanding of components in the whole workflow. This section focuses on identifying and introducing individual components in the workflow. Whereas, technical details and steps of implementation will be discussed in section 4.5.

4.4.1 Flow Breakdown

4.4.1.1 Input Event File

Two types of relationships are extracted from the input event file: DDI and DPI. For molecular and gene ontology matching, we require a pair of drugs whose separate ontologies were calculated first and then compared together for similarity. There are very few DDI and DPI drug pairs listed within the event file. If ontologies of the listed drug pairs are matched only then they create very sparsely joint clusters. To take advantage of the clustering algorithms we need a method that would create more connected clusters. For this purpose, rather than focusing on drug pairs in DDI, we make use of all unique drugs from DDI and DPI. A combined data set is created for all unique drugs in DDI/DPI called [Drug Data Set A]. The data set is the input for both Molecular Ontology Matching and Gene Ontology Matching.

4.4.1.2 Molecular Ontology Matching

Ontologies of each individual drug are calculated using ChEBI database to a tree depth of three levels as mentioned earlier. The structural information is discarded and only terms in the ontological tree are stored out corresponding to each drug. Molecular similarity is calculated by matching ontologies of every drug with all the drugs in the drug data set. The

count of common ontology terms in ontologies of two drugs that are being compared is calculated. The value of count represents a scalar measure of molecular similarity between drugs. Result of this stage is output file containing drug pairs and their respective count of matches in their ontologies.

4.4.1.3 Gene Ontology Matching

In calculating gene ontology matching, only DPI relationships from the event file are used as input to this stage. First step is to extract gene products for all proteins in the DPI using the GO database and then calculate gene ontology up to a depth of three levels for all the gene products. Finally, these gene ontologies are added together against its corresponding drug.

Each drug may correspond to one or more proteins. Each protein may refer to one or more gene products and each gene product has its own ontology tree as shown in figure 1. All the relationships between drug and ontologies are then collapsed. Each drug now corresponds to a combined gene ontology that is derived from gene products, which are derived from proteins belonging to the drug.

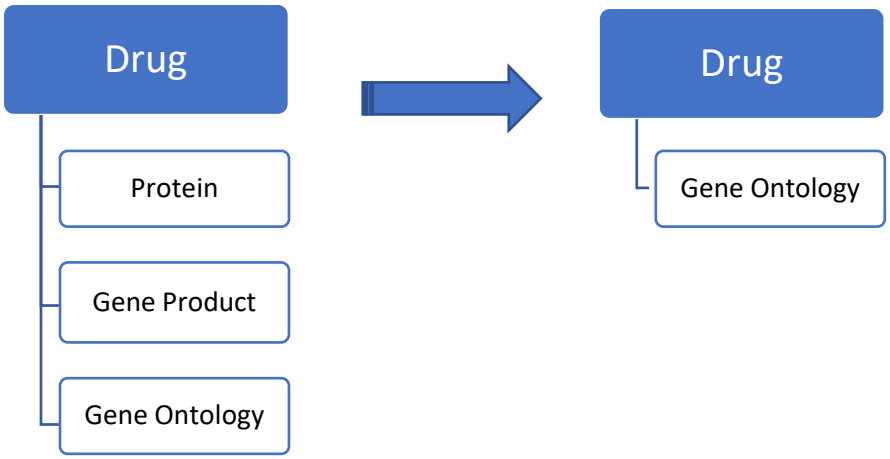


Figure 1: Combining hierarchy of relationship between drug entities.

Once we have corresponding gene ontologies for all drugs, next step is to compare them together. Each drug gene ontology is compared with all other drug ontologies. Results are listed as drug pair and weight that corresponds to number of terms that were common to both the drug's gene ontologies.

4.4.1.4 Normalisation

At this stage, we have two data sets: one is output of molecular matching and the other is output of gene matching. To carry out any comparison between the two, we need to first normalise both the data sets although molecular matching and gene matching are features of the drug but they have different scales i.e. they have different maximum and minimum values. To combine the ontologies of drug pairs together, equalisation of scale difference is necessary which is done by normalizing the data set. Normalisation is a straightforward method; dividing column with maximum weight. After division, the scales are balanced as the new calculated value represents a ratio. The weight varies between 0 and 1 which is a very small range and to make calculation easier, it is multiplied with 100. This updated range varies from 0 to 100. Normalization is carried out to the weights of both molecular and gene weights.

4.4.1.5 Thresholding

Next step is to carryout thresholding as most of the weights are insignificant; which if used in plotting data distribution, will result in a very strong positively skewed data. This means there would be many drug pairs with very small weights which do not have much effect. These weights are dropped due to their minimal effect. Data distribution will always remain right/positively skewed but better threshold means 1st and 3rd quarter is spread across the data. Thresholding value is chosen after performing experiments to find the ideal threshold. Its found to be 10% of normalised weights for both drug pairs of molecular and gene ontology.

4.4.1.6 Combining Ontologies

Now, there are multiple options available; we can either combine molecular and gene ontology or cluster them separately. Furthermore, both can be combined in two different ways either combining exact drug pairs in both ontologies or combining all drug pairs in both ontologies. Multiple iterations of the experiments are performed exploring these options.

4.4.1.7 Clustering & Ranking

Once the data set is finalised, it is loaded in GEPHI where clustering is carried out based on Chinese Whispers algorithm, Modularity with Resolution 1 and Modularity with Resolution 0.6. There are a couple of layout options to visually display the clusters. Ranking of drugs is also done across 5 centrality parameters for the whole network and individual clusters.

4.4.2 Iteration of Experiments

Iteration	Drug Data Set Used	Type of Join
First	Drug Data Set A	Inner
Second	Drug Data Set A	Outer

Table 7: Highlights relationship between Iteration, Drug Data Set and Type of Join.

First and Second Iterations are the obvious choices and they contain sum of both DDI and DPI interactions from the input files. First iteration only takes into account drug pairs that are common to both Molecular and Gene ontologies. As these drug pairs show some level of similarity across two ontologies, it is expected that the graph would have dense structure. Densely connected structures have optimized small world property, thereby it is expected that the output clusters would portray clear distinctions between them. Second Iteration is done with an outer join; it contains all the drug pairs from first iteration plus also the drugs that are not common to both ontologies but show a high normalised weight of above 50%. Considering drugs with significant high normalised weight if they are not common to both ontologies ensures that the added drug pairs are reliable and will not result in adding noise to overall clusters.

It is an experimental approach and nothing similar has been done in the past. Results of all these experiments will be evaluated by a biomedical domain expert to manually check if clusters were able to extract any meaningful division between drugs.

4.5 Implementation

Individual components of the experiment are listed below.

1. Data Preparation
2. Molecular Ontology Matching
3. Gene Ontology Matching
4. Normalisation
5. Thresholding
6. Combining Ontologies
7. Clustering & Ranking

Figure 2 highlights flow of important steps with respect to each component. Details of individual components are discussed in subsequent sections.

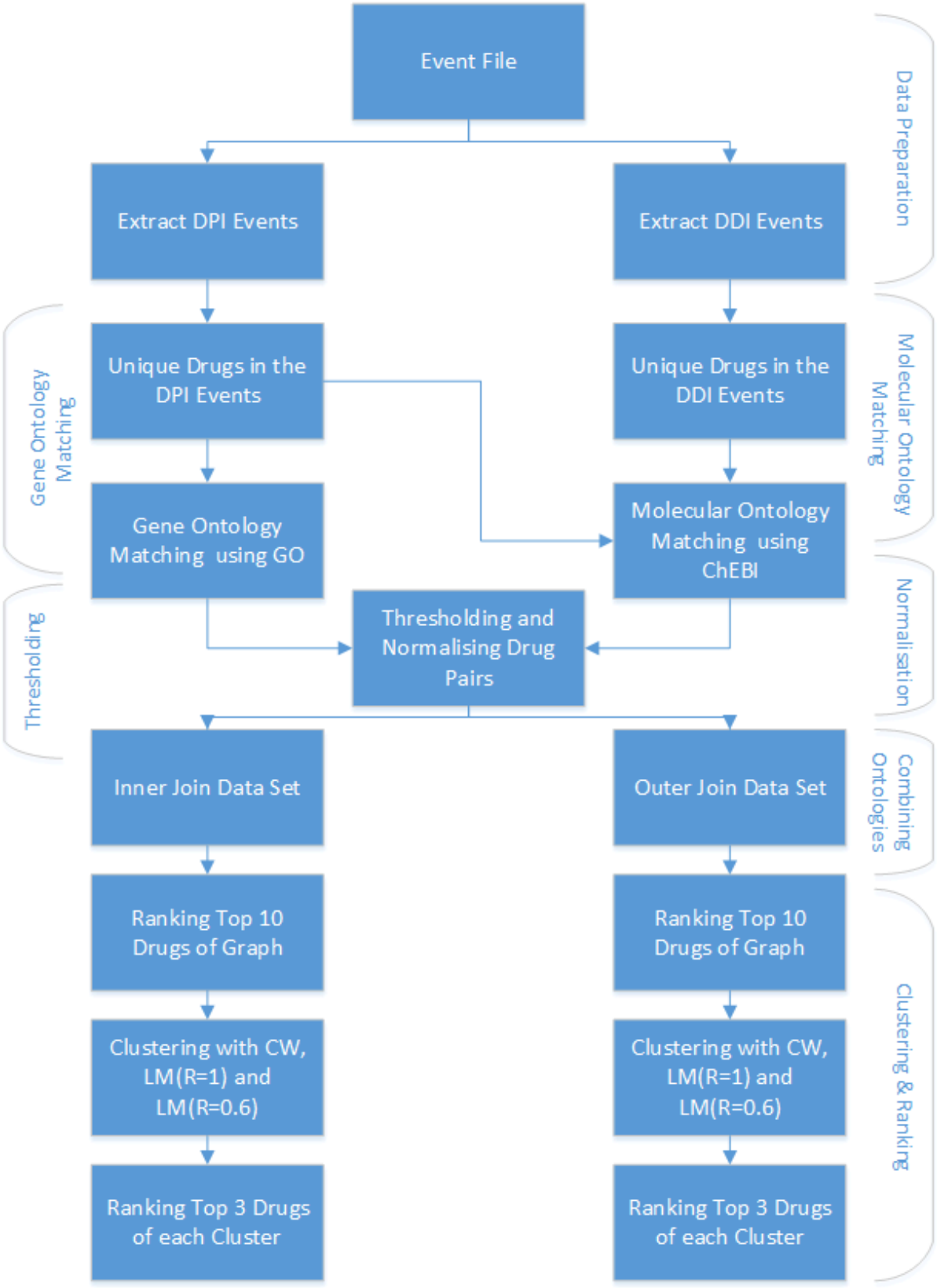


Figure 2: Flow of experiment.

4.5.1 Data Preparation

As a standard, .csv comma delimited file will be used for loading, exporting and storing of data. It is a lighter format which makes loading and reading of files easier for other software/tools.

4.5.1.1 Event File:

1. The event file is available in .tsv file format; first step is to change the file format to .csv. This can be easily done by opening a new excel file, going to data tab and selecting “From Text/CSV” file and giving path to .tsv file. Once the import is complete just save as .csv.
2. Before loading the data file into Python, we need to drop extra columns that are not required in this research. Drop all columns except Event ID, Primary Argument Ground ID, Secondary Argument Ground ID and Negation.
3. .csv files are loaded into Python Dataframes.
4. Protein to Protein Interactions are not used in the experiment and need to be dropped. It means that all tuples where Primary Argument Ground ID & Secondary Ground ID starts with “UniProt” are dropped.
5. For consistency, duplicates are checked and dropped for combination of Primary Argument ID and Secondary Ground ID.

4.5.1.2 Molecular Ontology:

1. “Flat File and Tab Delimited “database file is downloaded from ChEBI website.
2. Vertex and relation table are exported from .tsv file format to .csv.
3. Both Tables are loaded into Python Dataframes.

4.5.1.3 Gene Ontology:

1. SQL database dump is downloaded from GO website.
2. Database is loaded into a local server.
3. Three tables gene_product, dbxref and term2term are exported to.csv file format.
4. These files are then imported into Python Dataframes.

4.5.2 Molecular Ontology Matching:

Separate ontologies are calculated for each drug and then matched with all drug ontologies in Drug Data Set A.

4.5.2.1 Event File:

1. For both first and second iteration, unique drug ids are extracted from DDIs/DPIs in the event file by extracting all drugs where Primary Argument Ground ID is Drug and Secondary Argument Ground ID is Drug or either one of them is drug and other one is protein.
2. Next step is to calculate individual ontologies of drugs using ChEBI database. Each ChEBI Drug ID corresponds to a vertex in the ontology tree. Vertices combine together in a bottom up fashion to form an ontology tree. In the vertex table, vertex_ref column is searched for drug id and corresponding vertex id is noted.
3. This id is then searched in final_id of the relationship table. All corresponding init_id are parent nodes of that drug. This process marks completion of level 1 ontology matching.
4. For finding level 2 matching all the init_id found in level 1 matching are iteratively searched for their parent vertices. All parent ids in level 1 matching become child ids in level 2 tree ontology. So now, each child id is searched for parent id by searching for all init_ids that match the corresponding final_id.
5. The process is repeated one more time for level 3 matching. An example ontology tree for CHEBI: 10043 is shown in figure 2.

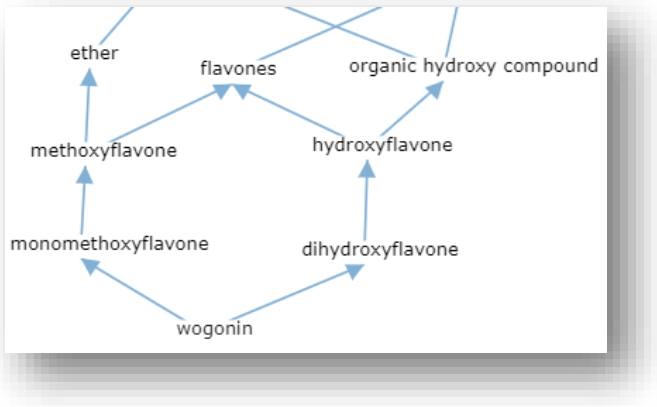


Figure 3: Ontology tree of level three depth for Wogonin drug [15].

6. The terms in the extracted tree are stored against their drug id; both structural information and depth information is discarded.
7. Once we have the ontologies of each drug, pair wise matching is done between ontologies of each drug with every drug in the data set. All common terms in ontologies of drugs being compared are considered as matches. Total count of the matches is said to be weight of similarity between the drug pair being matched.
8. Output of Molecular Ontology matching is a list of pairwise drugs with their corresponding weights.
9. When matching the ontologies of two drugs rather than just matching terms in the tree structure, their structural position is also considered i.e. term level in the tree is also

matched for similarity in the corresponding drug ontology tree, in this scenario the matching becomes too strict. An experiment was performed where structural position was also considered as a matching matrix but the results did not look very promising.

4.5.3 Gene Ontology Matching:

1. Extract all DPI relationships, all tuples where one of the primary or secondary ground id is drug id and other one is protein id.
2. For all proteins extracted from DPI, their corresponding gene products are found using GO database.
    a. Break protein ID into two parts. First part is always “UniProt:” and the second part contains the actual identifier ID.
    b. In dbxref table, search for id where xref_dbname equal ’UniProt’ and xref_key is equal to second part of protein id. This id is the unique identifier of the drug in GO database.
    c. Next step is to search for gene products for all proteins. ‘Type’ field in gene_product table represent individual term or node in the ontology tree. Gene_prodcut table is searched for type (Gene product id) corresponding to protein which is represented by dbxref key as found in the previous step. Each dbxref key (protein) can have one or more type (gene product).
3. Once we have all gene products against proteins, next step is to find ontologies of each gene product. Term2term table defines each edge as parent and child relationship. Term1 represent parent node and term2 represent edge node as described in data section.
    a. All gene products have their own ontology tree. Gene product ids found in step 2 are considered as base child nodes. For every gene product, their parents are found from gene_product table by finding term1 where term2 is gene product id. This is level 1 ontology.
    b. The process is repeated where all the term1 parent nodes in the previous part become term2 child nodes in this repetition to find another layer of ontology.
    c. The step is repeated again to find 3rd level of ontological terms.
    d. Similar to molecular ontology, all information about level and structural positioning of the tree is discarded, only terms are stored corresponding to gene product.
    e. Terms of Gene product ontologies are combined together based on gene products that correspond to the same protein.
    f. Ontological terms stored against protein are further combined based on protein correspondence to drugs. To summarise, multiple trees are combined based on their protein relation and further these relations are combined based on their drug associations. The final output is drugs and their ontological tree terms.
4. Next step is to carry out a comparison between drugs and their ontologies. Each Drug ontology is pairwise matched with all other drug ontologies. The count of number of similar terms in their ontologies is stored as weight, a measure of similarity between two drugs based on their gene ontology. The output of this step is pair of drug list and their corresponding weights.

4.5.4 Normalisation

In the First and Second iteration, we want to combine the weights of gene and molecular ontology corresponding to drug pairs. This is not directly possible as these are two different features of drugs and naturally have different weight ranges. Normalisation is done to standardise data values from a dynamic range to a specific range [70]. Therefore, normalisation is done on both drug pair weights of both ontologies.

a. To carryout normalisation, maximum weight in both the data sets is chosen and whole data set is divided by it. Output is a ratio of weights varying from 0 to 1.
b. Then, to redistribute the ratio of weights over some range, all weights are multiplied by the highest value of range we want to distribute it over. In this case, weight range is chosen to be 0 to 100 which meant multiplying whole data set with 100.

4.5.5 Thresholding

1. Both data sets consist of pair of drugs correlating each drug with every other drug. After observing the data set, it can be clearly seen that most of the data sets have fairly small weights. These weights are very small compared to the rest of the Data Set And add negligible information. Connectivity is also one of the parameters that is considered when clusters are computed for both Chinese Whispers and Louvain Modularity. To avoid these small weights to overpower cluster connectivity, these weights are removed.
    a. Maximum normalised weight varies from 0 to 100; a threshold limit is placed over lower 10% of the data set range which reduces this to 10 to 100.

4.5.6 Combining Ontologies

1. For the First iteration, we perform an inner join between drug pairs of two ontologies. Inner Join means drug pairs that are common to both molecular and gene data sets are retained and their weights averaged.
2. For the second iteration, outer join is applied for combining drug pairs of two ontologies together. Outer Join means all drug pairs belonging to molecular and gene data sets are retained. Drug pairs which are common to both are combined but drugs which belong to only one of the two data sets are penalised with a minimum threshold of 50% to ensure consistency of the data set.

Ontologies are combined in different ways for different experiment iterations. Output is exported in .csv files.

4.5.7 Clustering & Ranking

For both iterations, a similar process is followed for doing clustering.

1. Data set is loaded in GEPHI via Data Laboratory.
2. "YifunHu" [30] is chosen as graph visualisation algorithm.
3. As a post data processing step, a filter is applied to filter out scattered nodes. The filter removes nodes which are not part of the larger graph structure.
4. Nodes are ranked across 5 centrality parameters including Degree, Closeness, Betweenness, PageRank and Eigenvector centrality. Ranking of top 10 drugs is discussed in results section.
5. Three clusters are drawn for each iteration, Chinese Whispers, Modularity with Resolution 1 and Modularity with Resolution 0.6.
6. Drugs are ranked inside each cluster using the same centrality measures used to rank the whole network. Ranking of top 3 drugs is discussed in results section.
7. Steps 1-3 are repeated for first and second iteration.

4.6 Evaluation

Goal of clustering was to combine similar drugs together based on the objective function that maximises intra-cluster connectivity and minimise inter cluster connectivity with the underlying assumption being "higher the connectivity, similar the drugs".

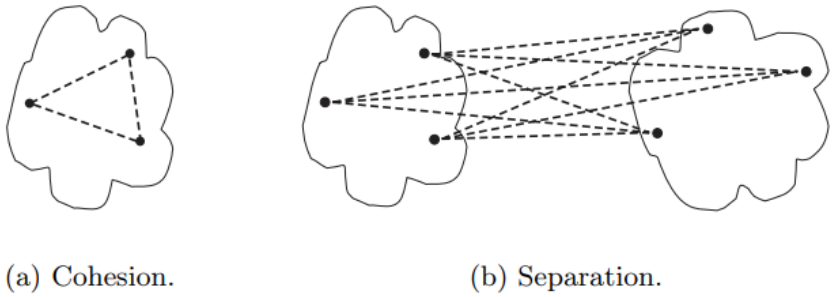
4.6.1 Internal Criterion:

The quality of intra and inter clusters is internal criterion of clustering [71]. There are different parameters available to measure this internal criterion such as modularity of clusters, Davies–Bouldin index, Dunn index, Silhouette coefficient etc. All these algorithms have a common goal to standardise the quality of clusters based on density of edges within clusters and their connecting edges. There are few problems and limitations of internal criterion

1. Problems: These internal criterions are biased if the mode optimizes the same parameter as the parameter being used to evaluate the cluster, the results would be correct from just one perspective. For example, if Louvain Modularity is used to draw communities and modularity is used as an internal criterion, the results would be biased; as modularity was the parameter that was being optimized by the algorithm.
2. Limitations: It has been observed that good internal criterion does not necessarily mean that the clustering results would be effective [71]. Effectiveness usually varies from case to case basis. In our case, high internal criterion does not necessarily mean that clusters represent similar drugs in terms of their dominant MOA.

4.6.1.1 Silhouette Coefficient

For evaluating clusters of different experiments, a parameter known as mean silhouette coefficient [72] is used. The technique was first proposed in 1987 and since then has been widely accepted as an internal criterion to judge quality of clusters where class labels are not known. Silhouette coefficient takes into account both "cohesion and separation" of a node. Cohesion focuses on the quality of inter clustering whereas separation focuses on the quality of intra clustering as shown in figure 4. The goal of good clustering is to have high separation value between clusters where each cluster also has high cohesion value. In this research, silhouette coefficient is used to validate clusters but it can also be used to find the optimized cluster. For example, clusters with low cohesion can be split for better clusters whereas, low separation and relatively similar cohesion clusters can be combined to form better clusters.



(a) Cohesion. (b) Separation.

Figure 4: Difference between cohesion and separation [73].

Let's discuss method to calculate Silhouette coefficient for each node.

1. Calculate the distance between node in observation with all other nodes of cluster as $\{a_i\}$.
2. Now, we compare the node in observation with all clusters that do not contain that node. Calculate average distance between node and all nodes of a given cluster. Iterate the process for all cluster to find the minimum average distance as $\{b_i\}$
3. For any object silhouette coefficient is represented by $s_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}$

The coefficient varies from -1 (minimum coefficients value) to 1 (maximum coefficients value). For the final value of coefficient, anything positive is considered good clustering and anything less than zero is not desirable. $\{a_i\}$ should be less than $\{b_i\}$ for good clustering and ideally it should be close to zero. Average (mean) Silhouette coefficients of all nodes inside a graph is calculated for clusters produced by different algorithms. The mean silhouette coefficient value represents overall cohesion and separation for clusters produced by a specific algorithm. Therefore, the coefficient value is used to compare quality of clusters in different algorithms.

Quality of an individual cluster can be measured by averaging silhouette coefficients of all nodes belonging to that cluster. To improve mean silhouette coefficient of the network where mean silhouette coefficient of network is particularly bad certain steps can be taken.

1. Mean silhouette of individual clusters is calculated.
2. Clusters that have poor silhouette coefficient are then dropped from the network. As these clusters add poorly to the mean silhouette coefficient of the network, dropping them increases silhouette value.

This is an extreme measure but necessary when clustering algorithm fail to place few nodes in correct clusters to optimize cohesion and separation. In certain scenarios, the problem can also arise due to limitations of the graph structure itself such as cases where some pairs of nodes are unconnected to the larger structure of the network.

The technique was employed to improve results of second iteration where the network had a lot of broken nodes, unconnected to the larger structure. As expected, these broken nodes were negatively affecting the silhouette value. A comparison of silhouette value of before and after dropping the nodes is also discussed in section 5.5.1.

4.6.2 External Criterion

Due to the problems and limitations of internal criterion, it is not the only way in which results are evaluated. To verify if the drugs share some biological identifiable property between groups, an external criterion is employed i.e. the clusters are manually checked by a biological domain expert. This is different from internal criterion which parametrised results based on inter and intra cluster quality whereas this criterion focuses on true natural distinction between nodes and clusters.

The ideal way to do it is the formation of gold standard clustering data set by more than one evaluator with significant high inter-judgement agreement.

1. Though, it is not straight forward as there is no fine line that distinguishes between drugs of different clusters. The gold standard may be designed focusing on therapeutic effects of drug but this is not ideal MOA which will be reflected by different communities; different groups may reflect different dominant MOA. So, formation of gold standard

clusters is out of question as a domain expert may know MOA but a drug may have more than one MOA and deciding dominant MOA of the drug is also a question. Furthermore, calculating similarity weights and observing patterns in connectivity of drugs to identify which drugs belong together is also not feasible.

Simplifying the graph structuring by ignoring weights of edges in the graph network and only considering connectivity of the graph structure as unweighted graph even for that scenario considering the size of the database and dense connectivity, it is not humanly possible to decide the property on which clusters would be formed before performing the experiment. Similarly, in this research, the only thing that can be stated certainty is that there will be some level of similarity between the clusters. The exact MOA that similarity highlights and level of similarity needs to be judged manually. Identifying and verifying MOA of clusters is not within the scope of this research.

2. If we still manage to make a gold standard with some overlaps, there also exists a problem of granularity. Some clusters would be split into further (two or more) clusters in original Data Set As compared to gold standard. On the other hand, some clusters would be combined together into one class as compared to gold standard where they might be represented in two or more classes. The clustering is not wrong but is a matter of granularity.

Having explained the two problems, the most viable option seems to be manually checking each cluster for consistency by a biomedical domain expert based on the property of similarity highlighted in the cluster. For this purpose, each cluster is divided into two classes; these are just nodes that are correctly classified (True Positive) and nodes that are wrongly classified (False Positive). Just highlighting true positive and false positive is not similar to doing comparison with gold standard data. While evaluating with gold standard data true class labels are known so result set is divided into 4 parameters: True Positive, True Negative, False Positive and False Negative. These values combine together to form a confusion matrix. Confusion matrix can make further evaluation comparatively easy, as multiple evaluation parameter can be evaluated on its basis including Rand measure, F-measure, Dice index and Fowlkes–Mallows index & Jaccard index.

Evaluation of clusters based on exterior criterion either against a gold standard (creating manually annotated class labels) or manually checking each node for belongingness to a specific cluster is a time-consuming process and requires expertise in biomedical domain. In our case, owing to time and resource limitations, each cluster was not separately analysed by the domain expert and feedback was only generic in nature after going over some specific drug clustering cases.

4.6.3 Comparison of Chinese Whispers Vs Louvain Modularity

To further strengthen the results, clusters of all algorithms were compared to each other in order to identify generic similarities. The purpose of comparing clusters is to ensure that the results are consistent to a certain degree of confidence across multiple algorithms. Expectation is that clusters should remain similar across algorithms with slight changes in boundary nodes, which is acceptable based on computational variations across algorithms. Another expected output is that clusters produced by different algorithms might represent super-cluster and sub-cluster relationships. The output of comparison would be a hierarchical breakdown of clusters across different algorithms; this is discussed in section 5.2.6 and 5.3.6.

Chapter 5: Result and Analysis

5.1 Overview

			Algorithms		
			Chinese Whisper (CW)	Louvain Modularity with Resolution 1 (LM[R=1])	Louvain Modularity with Resolution 0.6 (LM[R=0.6])
Iteration	First	Inner Join	1	2	3
	Second	Outer Join	4	5	6

Table 8: Shows experiment number corresponding to iteration and algorithm used.

A total of six experiment were performed. The experiments were broken down in two iterations; where each iteration was repeated using three algorithms: Chinese Whispers, Louvain Modularity with Resolution 1 (LM[R=1]) and Louvain Modularity with resolution 0.6 (LM[R=0.6]). Both iterations made use of Drug Data Set A as input data. The major difference between the two iterations was the procedure of combining molecular and gene ontologies. For first iteration only drug pairs that were common in molecular and gene ontology result-set are combined. For second iteration, all drug pairs in molecular and gene ontology result-set are combined with certain threshold restrictions. It includes drug pairs that were common to both ontologies and additional drug pairs which had considerable high weight but are not shared by both ontologies.

The result of each experiment i.e. clusters was represented in the form of graph using Yifan Hu Algorithm. The algorithm has been introduced earlier in the methodology section. For comparison purposes, each cluster will be assigned a cluster id. For each cluster, its cluster distribution will also be calculated to show cluster's coverage of the overall network. Cluster distribution is calculated by finding percentage of nodes inside the cluster with respect to total number of nodes in the graph structure.

5.1.1 Statistics of Data set

Serial	Characteristics of Network	Inner Join [First Iteration]	Outer Join Old [Second Iteration]
1	**Total Node**	43	89
2	**Nodes in Connected Structure**	43	72 (80.9% Visible)
3	**Total Edges**	145	260
4	**Edges in Connected Structure**	145	251 (96.54% Visible)
5	**Graph Density**	0.161	0.098
6	**Average Degree**	6.744	6.972
7	**Average Path Length**	2.205	2.712
8	**Average Clustering Coefficient**	0.712	0.699

Table 9: Statistic of connected graph structure in two iterations.

1. **Total Node:** Drugs/Vertex
2. **Nodes in Connected Structure:** Nodes which are part of the large connected structure in the network.
3. **Total Edges:** Relationship between Drugs/Vertices
4. **Edges in Connected Structure:** Edges which are part of the large connected structure in the network.
5. **Graph Density:** It represents the density of connectivity for the whole graph. The value varies from zero to one. One is the highest density achievable, in case where every node is connected with every other node. For undirected graphs, it is calculated by the formula: $Graph\ Density = \frac{2(No.of\ Edges)}{|No.of\ vertice|(No.\ of\ vertice-1)}$ [74]
6. **Average Degree:** It represents the average number of edges for each node in the network. Calculated by the formula.

$$Average\ Degree = \frac{(degree(node\ 1) + drgree(node\ 2) \dots degree(node\ 3))}{total\ no.of\ nodes}$$

Two data sets are chosen deliberately to ensure that they have an increasing number of drugs in each consecutive data set from left to right. Some important observation about the data set are listed:

- Inner join data set is small but it has a higher data density. This is consistent with the fact that inner join data set only contains drug pairs that are common to both drug and gene ontology. This means that each drug pair shows a high level of confidence in the similarity that they represent; as each drug pair has support of both molecular and gene ontology.
- Second data set is outer join which contains all the drug pairs from inner join data set. Additionally, it also contains drugs that are not common to both the ontologies but have

high weight associated with them. Drug pairs that are not shared by both ontologies have a threshold limit of 50% normalised weight. The addition of these drug pairs means more nodes in comparison to inner join as the criterion for the node to be added in data set is less sensitive. But the disadvantage of this flexibility is that it leads to addition of sparsely connected nodes. So, graph density of outer join is slightly less than that of inner join.

- Both Graph Density and Average Degree are directly correlated; higher the degree, higher the density if the number of nodes remains constant. Both these parameters have an impact on quality of clusters; very small graph density implies that there would be many small clusters. Comparatively very large cluster density would cause formation of network where every node is connected with every other node which is also not ideal as it would just result in one big cluster.
- Average clustering coefficient and average path length combined portray measure of "small world property". Higher the small world property, better the result of clustering for both algorithms. However, in this case both data sets share the same moderate path length and clustering coefficient which implies that both data sets share the same small world property.

5.2 First Iteration

5.2.1 Centrality Parameter Distribution

Appended below are the distributions of Degree, Betweenness, Closeness, Page Rank and Eigenvector Centrality for network of first iteration.

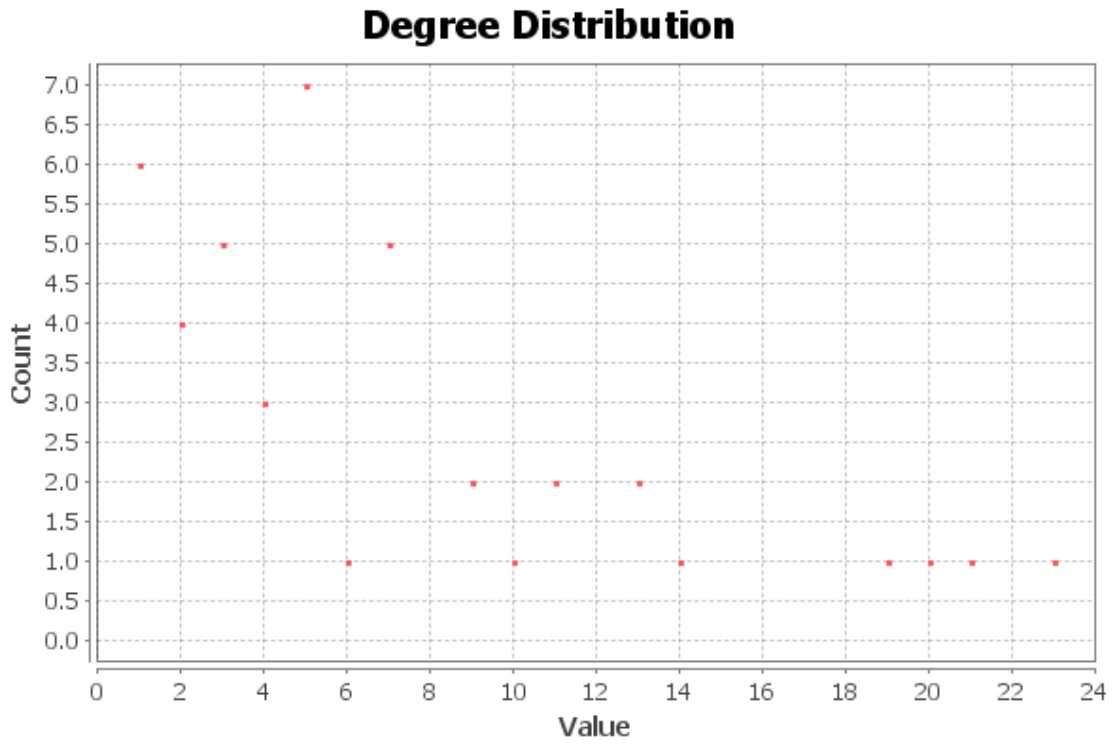


Figure 5: Degree distribution of network in first iteration.

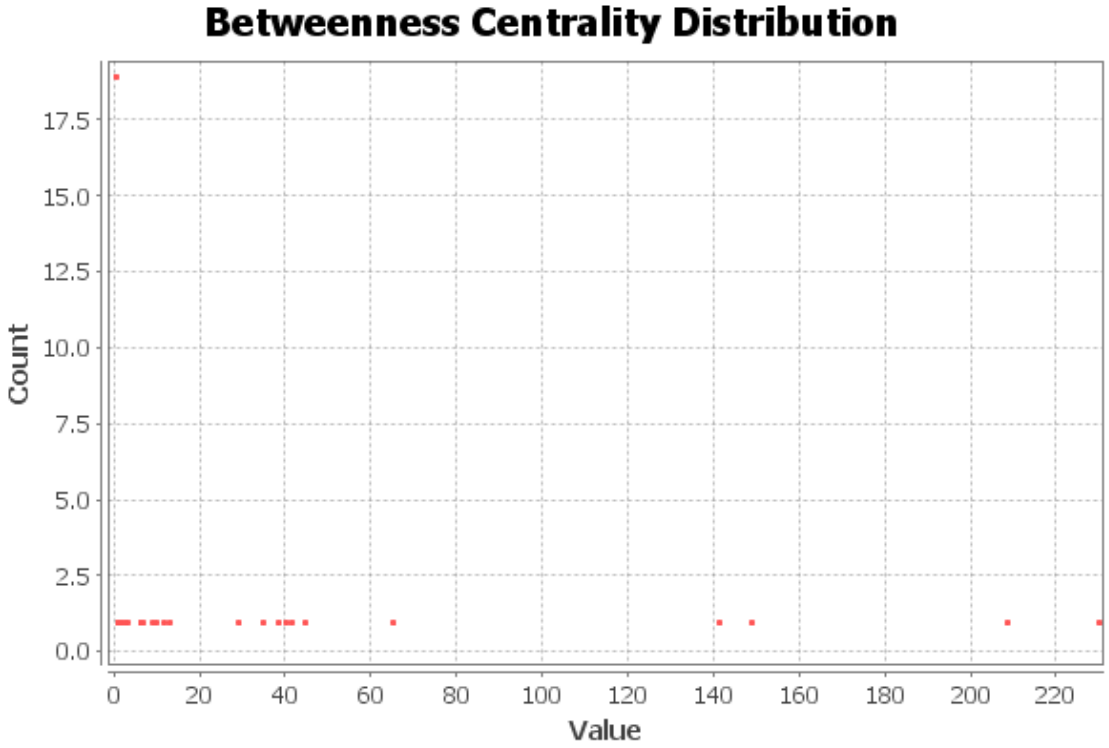


Figure 6: Betweenness distribution of network in first iteration.

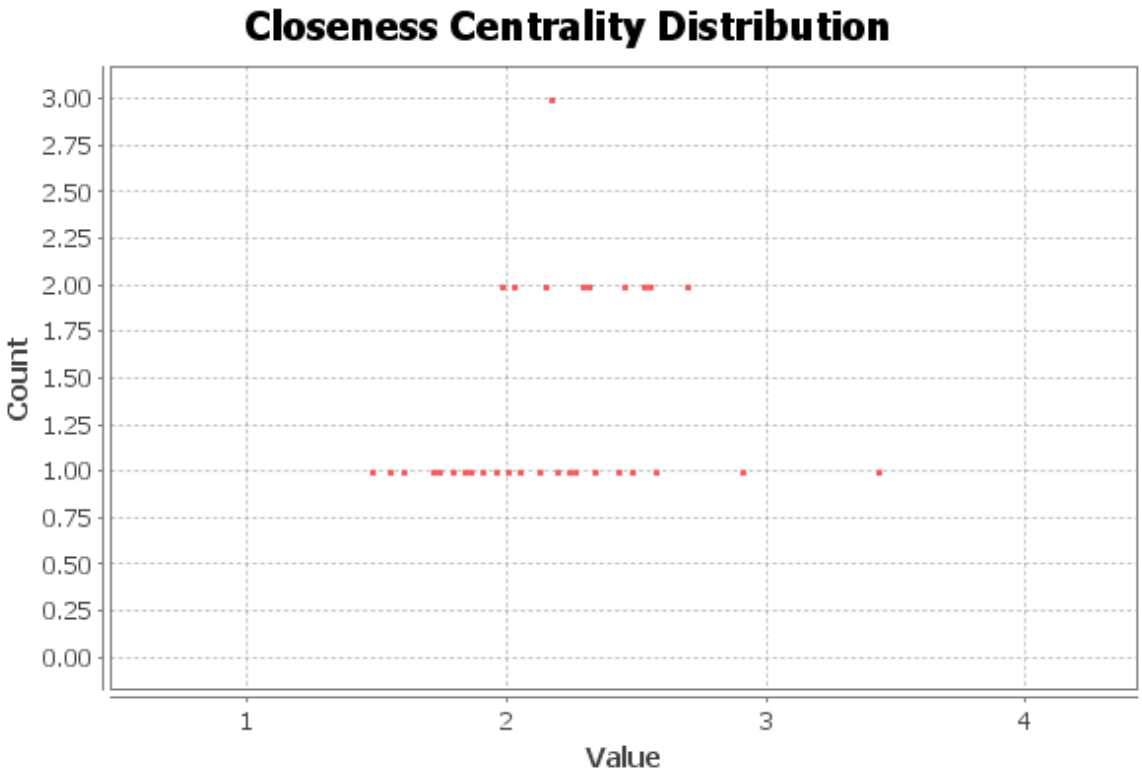


Figure 7: Closeness distribution of network in first iteration.

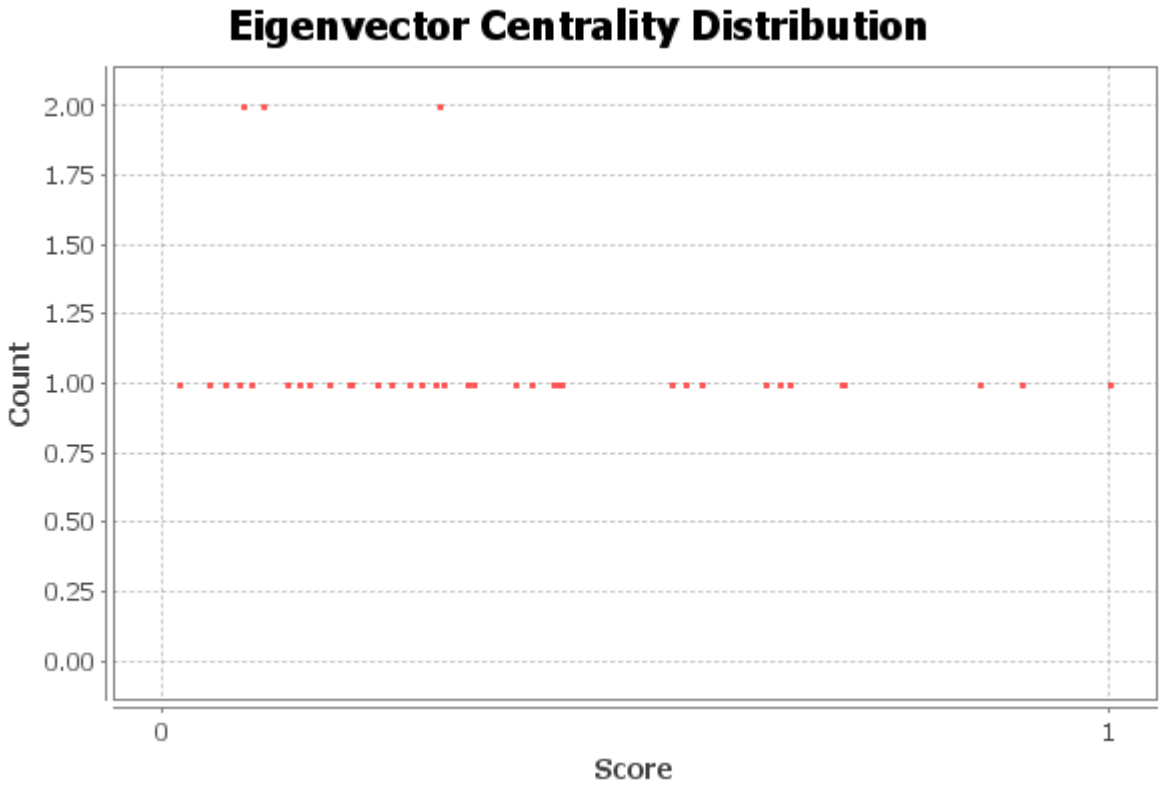


Figure 8: Eigenvector distribution of network in first iteration.

5.2.2 Drug Ranks Via Centrality

Top 10 drugs across 5 centrality parameters were calculated for the whole network. Results are shown in table below:

S#	Label	Degree	Label	Closeness Centrality	Label	Betweenness Centrality	Label	PageRank	Label	Eigenvector Centrality
1	sirolimus	23	tyrosine	3.43	L-threonine	229.71	L-threonine	0.074	sirolimus	1.00
2	sorafenib	21	sphingosine 1-phosphate	2.90	sirolimus	208.06	sorafenib	0.072	sorafenib	0.91
3	curcumin	20	(-)-cubebin	2.69	curcumin	148.56	sirolimus	0.070	curcumin	0.86
4	L-threonine	19	cabozantinib	2.69	sorafenib	141.01	curcumin	0.066	vemurafenib	0.72
5	vemurafenib	14	bryostatin 1	2.57	L-serine	64.97	L-serine	0.058	selumetinib	0.72
6	selumetinib	13	prostaglandin E2	2.55	hydrogen peroxide	44.32	selumetinib	0.044	L-threonine	0.66
7	L-serine	13	cholesterol	2.55	L-threonine residue	41.00	vemurafenib	0.042	trametinib	0.65
8	trametinib	11	dihydroxyacetone	2.52	bortezomib	39.82	PLX-4720	0.035	PLX-4720	0.64
9	PLX-4720	11	trans-resveratrol	2.52	trametinib	37.98	trametinib	0.033	lenvatinib	0.57
10	U0126	10	3',5'-cyclic UMP	2.48	selumetinib	34.46	lenvatinib	0.032	U0126	0.55

Table 10: Top 10 drugs with respect to five centrality parameters in first iteration.

5.2.3 First Iteration & Chinese Whispers- Experiment 1

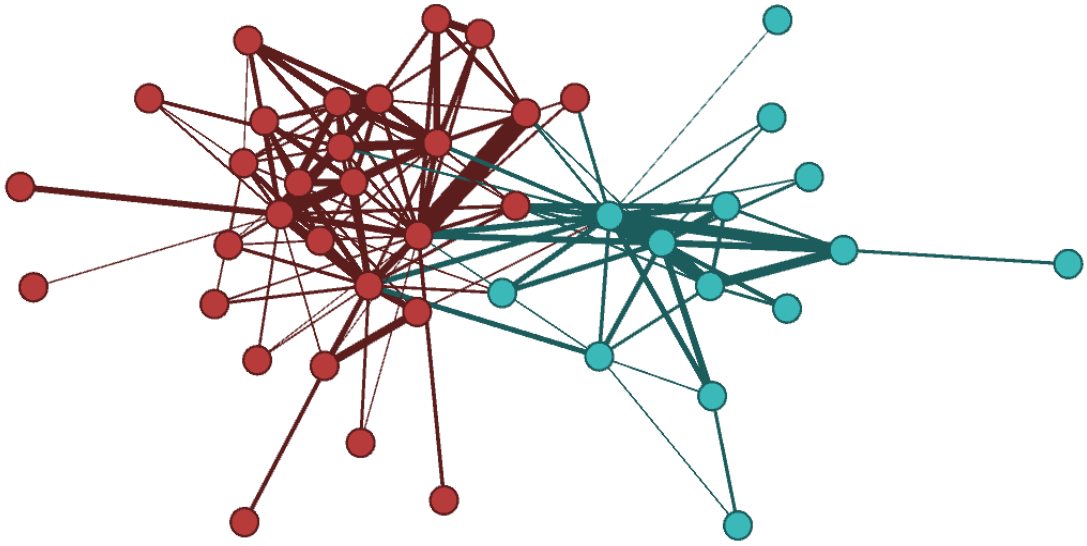


Figure 9: Yifan Hu representation for clusters in 1st iteration using CW algorithm.

5.2.3.1 Cluster Distribution:

Cluster ID	Colour	Distribution
CW_C1		67.44
CW_C2		32.56

Table 11: Clusters in experiment 1.

- Outcome of first iteration is two large clusters identified using CW as shown in figure 9. Yifan Hu graph representation algorithm is used to plot the graph.
- In Table 11 a cluster identification is assigned to each cluster, colour of each cluster is also shown corresponding to figure 9 and comparison of cluster size is also done by calculating node distribution across clusters.

5.2.3.2 Drug Raking in Clusters

Cluster ID	Rank	Degree	Closeness Centrality	Betweenness Centrality	PageRank	Eigenvector Centrality
CW_C1	1	sirolimus	(-)-cubebin	sirolimus	sorafenib	sirolimus
	2	sorafenib	cabozantinib	curcumin	sirolimus	sorafenib
	3	curcumin	trans-resveratrol	sorafenib	curcumin	curcumin
CW_C2	1	L-threonine	tyrosine	L-threonine	L-threonine	L-threonine
	2	L-serine	sphingosine 1-phosphate	L-serine	L-serine	L-serine
	3	hydrogen peroxide	bryostatin 1	hydrogen peroxide	L-serine residue	hydrogen peroxide

Table 12: Top 3 drugs in each cluster for experiment 1.

5.2.4 First Iteration & Louvain Modularity (Resolution 1)- Experiment 2

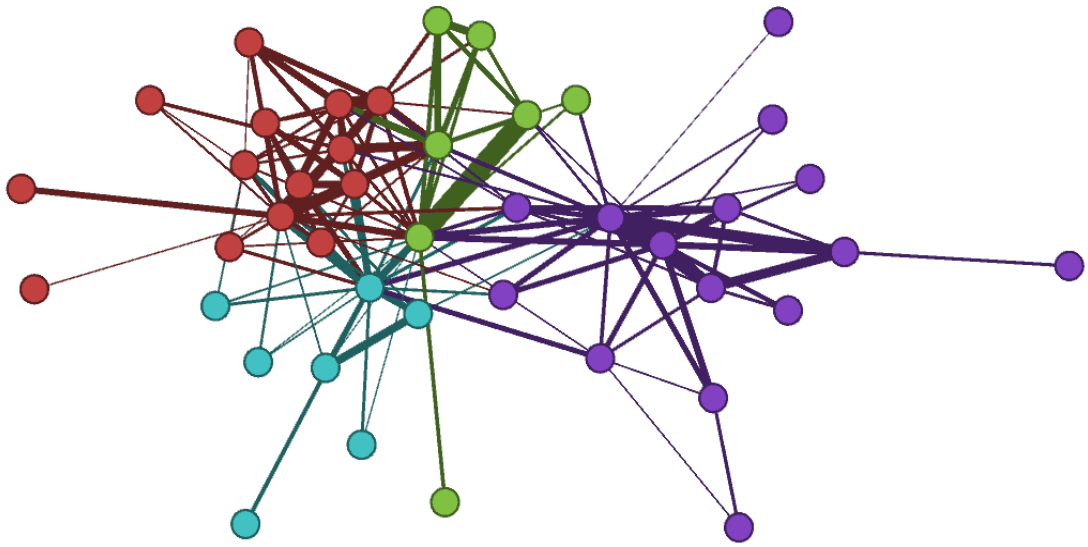


Figure 10: Yifan Hu representation for clusters in 1st iteration using LM[R=1] algorithm.

5.2.4.1 Cluster Distribution:

Cluster ID	Colour	Distribution
LM_C1		34.88
LM_C2		32.56
LM_C3		16.28
LM_C4		16.28

Table 13: Clusters in experiment 2.

- First iteration using LM[R=1] algorithm gives four clusters as shown in figure 10.
- In Table 13 a cluster identification is assigned to each cluster, colour of each cluster is also shown corresponding to figure 10 and comparison of cluster size is also done by calculating node distribution across clusters.
- Total modularity of the graph is 0.37

5.2.4.2 Drug Raking in Clusters

Cluster ID	Degree	Closeness Centrality	Betweenness Centrality	PageRank	Eigenvector Centrality
LM_C1	L-threonine	tyrosine	L-threonine	L-threonine	L-threonine
	L-serine	sphingosine 1-phosphate	L-serine	L-serine	bortezomib
	bortezomib	bryostatin 1	hydrogen peroxide	L-serine residue	L-serine
LM_C2	sorafenib	(-)-cubebin	sorafenib	sorafenib	sorafenib
	vemurafenib	cabozantinib	trametinib	vemurafenib	vemurafenib
	trametinib	2-(2-amino-3-methoxyphenyl)chromen-4-one	vemurafenib	PLX-4720	trametinib
LM_C3	sirolimus	doxycycline	sirolimus	sirolimus	sirolimus
	selumetinib	tandutinib	selumetinib	selumetinib	selumetinib
	everolimus	torkinib	everolimus	everolimus	everolimus
LM_C4	curcumin	trans-resveratrol	curcumin	curcumin	curcumin
	acetylsalicylic acid	dimethyl sulfoxide	acetylsalicylic acid	acetylsalicylic acid	acetylsalicylic acid
	paracetamol	cisplatin	paracetamol	paracetamol	paracetamol

Table 14: Top 3 drugs in each cluster for experiment 2.

5.2.5 First Iteration & Louvain Modularity (Resolution 0.6)- Experiment 3

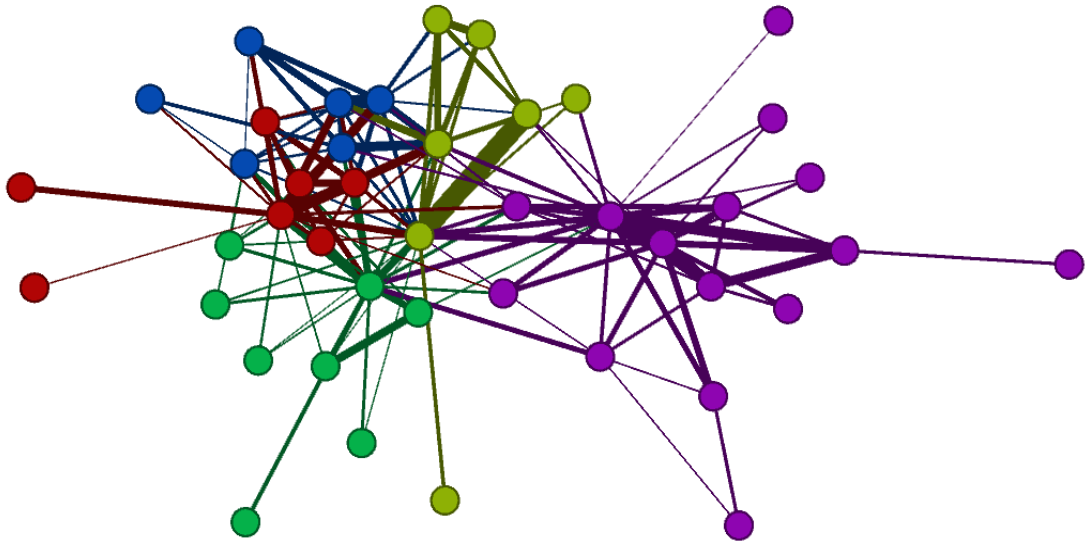


Figure 11: Yifan Hu representation for clusters in 1st iteration using LM[R=0.6] algorithm.

5.2.5.1 Cluster Distribution:

Cluster ID	Colour	Distribution
LM0.6_C1		34.88
LM0.6_C2		18.6
LM0.6_C3		16.28
LM0.6_C4		16.286
LM0.6_C4		13.95

Table 15: Clusters in experiment 3.

- Using LM[R=0.6] produces five clusters as shown in figure 11.
- In Table 15 a cluster identification is assigned to each cluster, colour of each cluster is also shown corresponding to figure 11 and comparison of cluster size is also done by calculating node distribution across clusters.
- Total modularity of the graph is 0.36.

5.2.5.2 Drug Raking in Clusters

Cluster ID	Degree	Closeness Centrality	Betweenness Centrality	PageRank	Eigenvector Centrality
LM0.6_C1	L-threonine	Tyrosine	L-threonine	L-threonine	L-threonine
	L-serine	sphingosine 1-phosphate	L-serine	L-serine	bortezomib
	bortezomib	bryostatin 1	hydrogen peroxide	L-serine residue	L-serine
LM0.6_C2	curcumin	trans-resveratrol	curcumin	curcumin	curcumin
	acetylsalicylic acid	dimethyl sulfoxide	acetylsalicylic acid	acetylsalicylic acid	genistein
	genistein	Cisplatin	paracetamol	paracetamol	acetylsalicylic acid
LM0.6_C3	sirolimus	Doxycycline	sirolimus	sirolimus	sirolimus
	selumetinib	Tandutinib	selumetinib	selumetinib	selumetinib
	everolimus	Torkinib	everolimus	everolimus	everolimus
LM0.6_C4	sorafenib	(-)-cubebin	sorafenib	sorafenib	sorafenib
	lenvatinib	Cabozantinib	lenvatinib	lenvatinib	lenvatinib
	sunitinib	Sunitinib	sunitinib	sunitinib	axitinib
LM0.6_C5	vemurafenib	2-(2-amino-3-methoxyphenyl) chromen-4-one	trametinib	vemurafenib	vemurafenib
	trametinib	Dabrafenib	vemurafenib	PLX-4720	trametinib
	PLX-4720	U0126	U0126	trametinib	PLX-4720

Table 16: Top 3 drugs in each cluster for experiment 3.

5.2.6 Analysis

The results were surprisingly accurate due to the fact that clustering was almost consistent for all three algorithms. Notable observations are:

1. Number of clusters increased slightly in each consecutive experiment starting from CW, LM1 to LM0.6. CW showed two clusters, LM[R=1] showed four clusters and LM[R=0.6] showed five clusters.
2. The modularity of the graph is the parameter being optimised by both LM[R=1] and LM[R=0.6] algorithms. Therefore, modularity of end clusters is a biased evaluation parameter and cannot be used in evaluation phase. Though, it is important to mention that both LM[R=1] and LM[R=0.6] maintain a high degree of modularity in their clusters.
3. Clusters show a behaviour consistent with the concept of super cluster and sub-clustering. All algorithms consistently produced more fine-grained (sub-clusters) version of clusters (super-clusters) produced by their predecessor algorithm.

CW produces only two super-clusters, whereas LM[R=1] produces four clusters and LM[R=0.6] breaks the network down to five clusters. Figure 12 shows the breakdown of clusters produced by the three algorithms in first iteration.

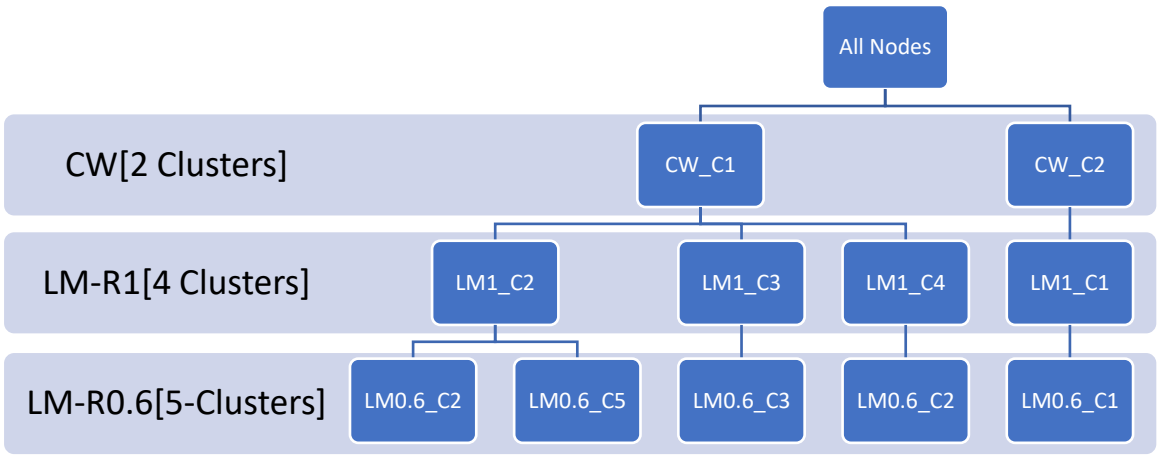


Figure 12: Relationship between cluster of different algorithms in first iteration.

The hierarchical structure between clusters can be explained by the fact that both Graph Density, Edge Weights or graph connectivity structure do not change by applying different clustering algorithms. The algorithms might perform differently such as choosing more sub-clusters based on algorithms internal cut-off and accuracy but base partitions should show observable relationships between clusters of different algorithms. Second reason for

breakdown to be so ideal is high graph density and average path length, both Louvain Modularity and CW give optimized results. While considering the ranking of top 10 drugs across centralities for the whole network, 6 similarities across 4 centrality measures of ranking were discovered. Sirolimus, sorafenib, curcumin, L-threonine and selumetinib drugs were common across Degree, Betweenness, PageRank and Eigenvector centrality.

Top 3 clusters across 5 centrality parameters were compared with analogous clusters (as identified by figure 12) across different algorithms. Following results were found:

i. Top 3 drugs across 5 parameters in CW_C2, LM_C1 and LM0.6_C1 were the same.
ii. Top 3 drugs in LM1_C3 and LM0.6_C3 clusters were also exactly the same.
iii. Top 3 drugs in LM1_C4 and LM0.6_C2 had just one dissimilarity in eigenvector centrality and one in degree centrality.

As the composition and size of related clusters across algorithms was relatively same due to that the centrality measures across these related clusters also remain the same.

5.3 Second Iteration

5.3.1 Centrality Parameter Distribution

Attached below are the distributions of Degree, Betweenness, Closeness, Page Rank and Eigenvector Centrality for network of second iteration.

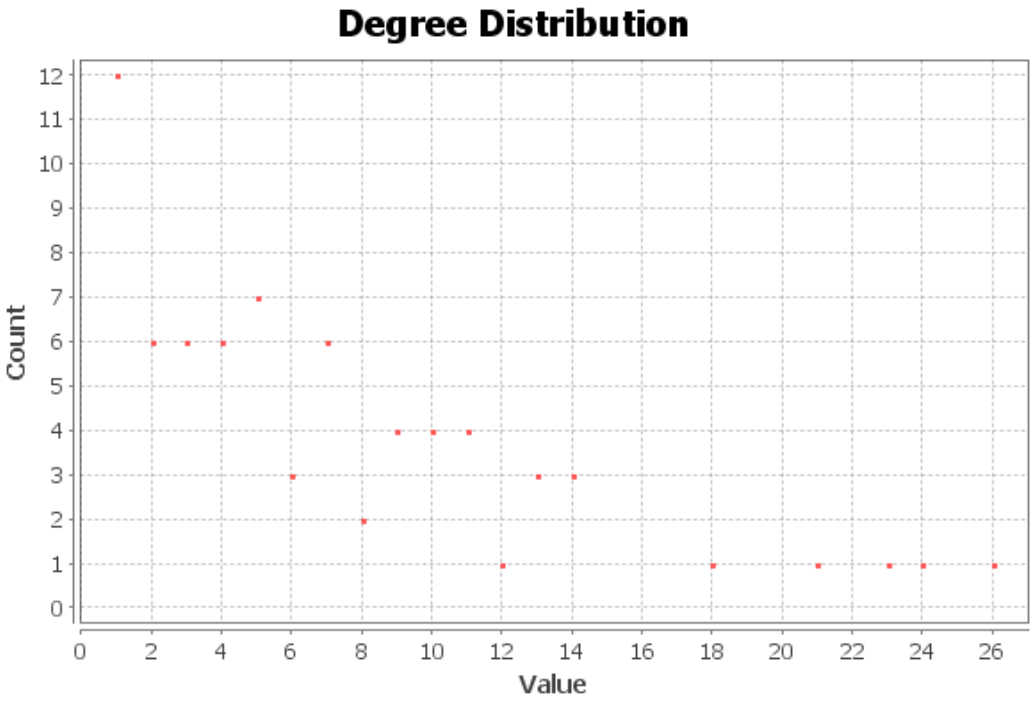


Figure 13: Degree distribution of network in second iteration.

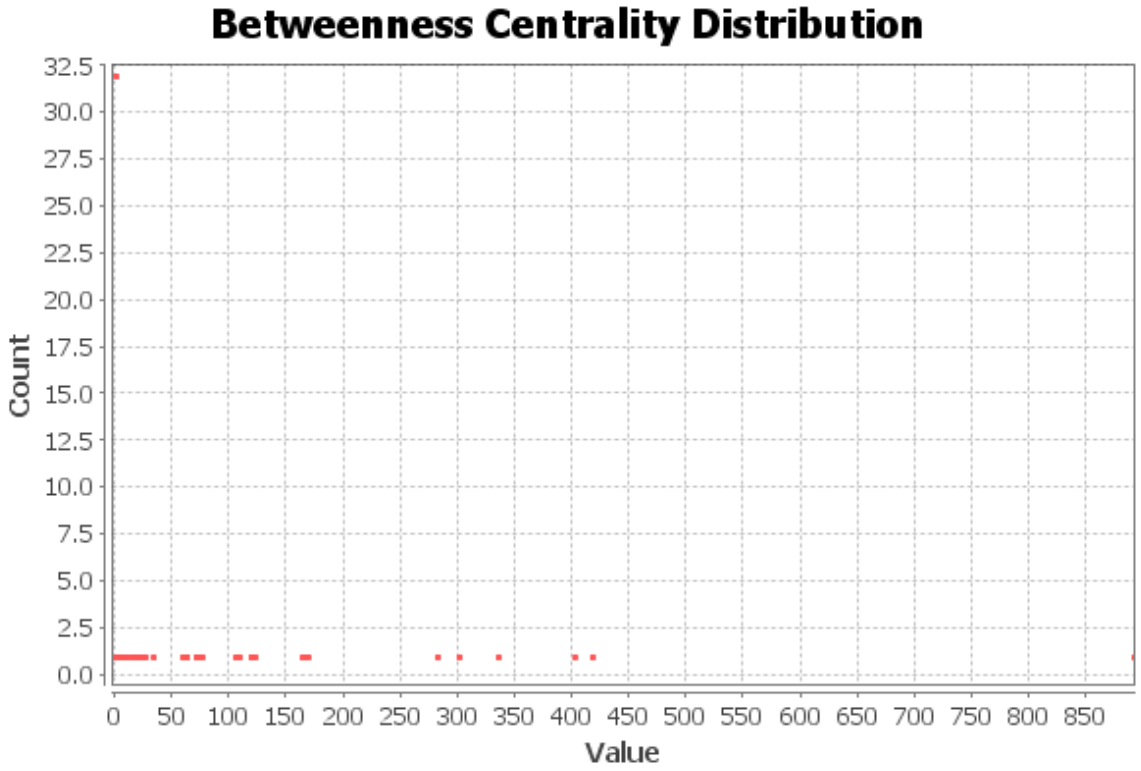


Figure 14: Betweenness distribution of the network in second iteration.

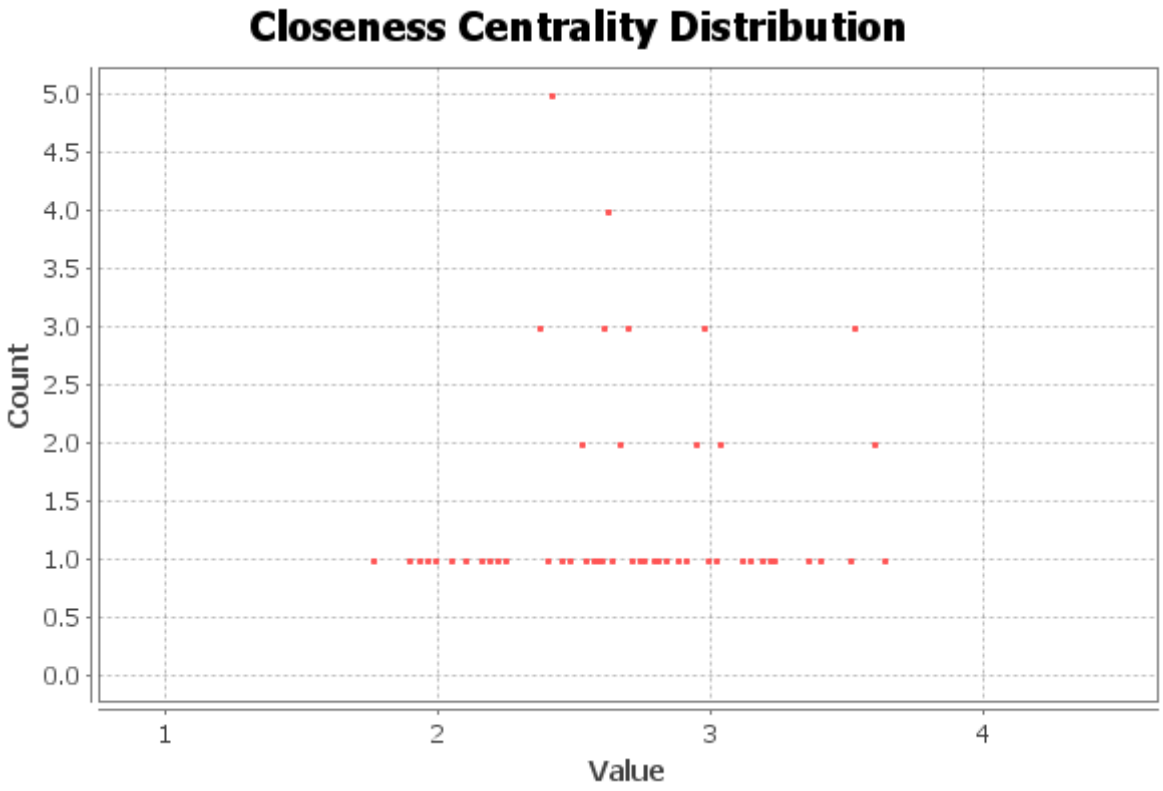


Figure 15: Closeness distribution of the network in second iteration.

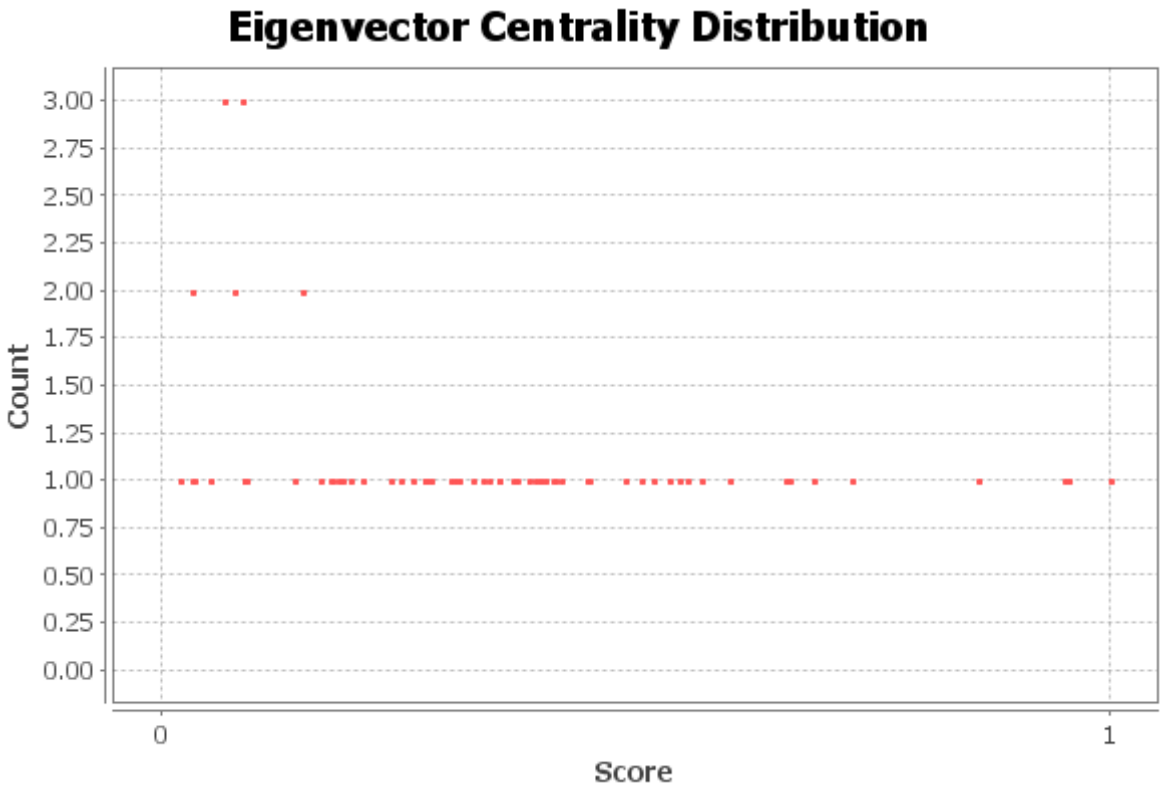


Figure 16: Eigenvector distribution of the network in second iteration.

5.3.2 Drug Ranks Via Centrality

Top 10 drugs across 5 centrality parameters were calculated for the whole network. Results are shown in table below:

S#	Label	Degree	Label	Closeness Centrality	Label	Betweennes s Centrality	Label	PageRan k	Label	Eigenvect or Centrality
1	L-threonine	26	L-gamma-glutamyl-L-cysteine	3.63	L-threonine	890.65	L-threonine	0.052	sirolimus	1.00
2	sorafenib	24	linifanib	3.59	sirolimus	416.51	Sorafenib	0.050	L-threonine	0.96
3	sirolimus	23	PD-153035	3.59	sorafenib	400.83	L-serine	0.041	sorafenib	0.95
4	curcumin	21	ibrutinib	3.52	curcumin	333.91	Curcumin	0.040	curcumin	0.86
5	L-serine	18	ceritinib	3.52	L-serine	300.38	Sirolimus	0.039	selumetinib	0.73
6	vemurafenib	14	GSK690693	3.52	tandutinib	280.78	tandutinib	0.029	vemurafenib	0.69
7	Glu-Phe-Val	14	desmosine	3.51	L-phenylalanine	167.38	selumetinib	0.027	L-serine	0.66
8	selumetinib	14	L-seryl group	3.39	selumetinib	164.15	Glu-Phe-Val	0.026	trametinib	0.66
9	Phe-Asn	13	L-tyrosine	3.35	ombitasvir	162.31	vemurafenib	0.024	PLX-4720	0.60
10	Thr-Ser	13	Glu-Met	3.23	trametinib	121.62	Phe-Asn	0.024	bortezomib	0.57

Table 17: Top 10 drugs with respect to five centrality parameters in second iteration.

5.3.3 Second Iteration & Chinese Whispers- Experiment 4

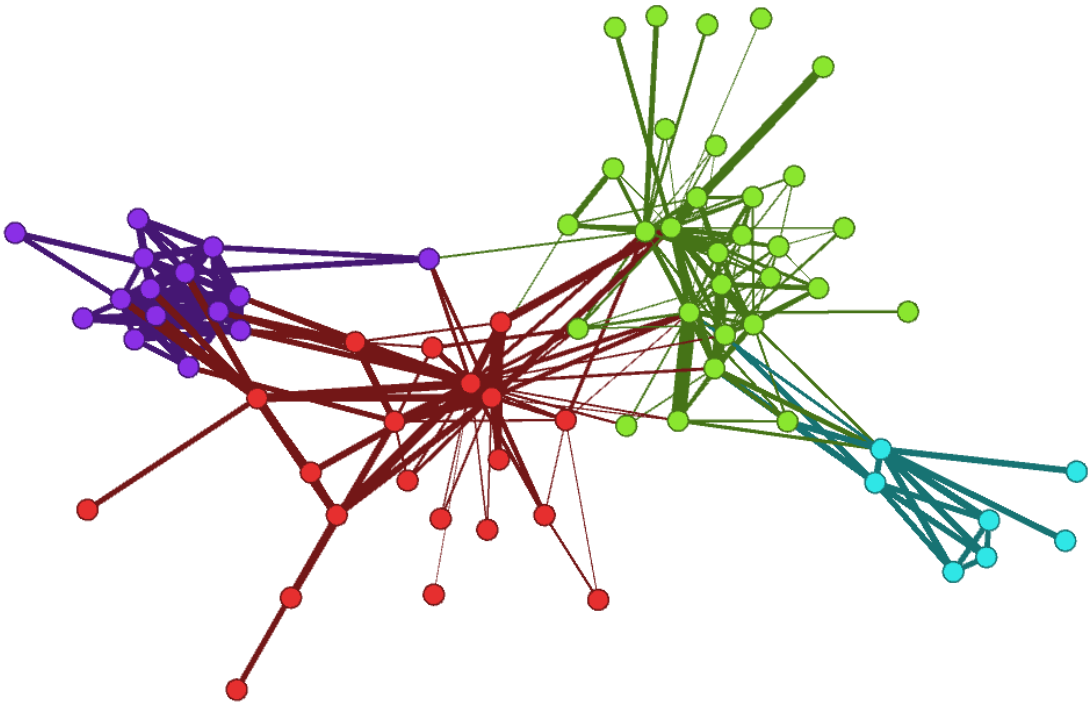


Figure 17: Yifan Hu representation for clusters in 2nd iteration using CW algorithm.

5.3.3.1 Cluster Distribution:

Cluster ID	Colour	Distribution
CW_C1		41.67
CW_C2		27.78
CW_C3		20.83
CW_C4		9.72

Table 18: Clusters in experiment 4.

- CW identified four clusters as shown in figure 17 drawn using Yifan Hu graph representation algorithm.
- In Table 18 a cluster identification is assigned to each cluster, colour of each cluster is also shown corresponding to figure 17 and comparison of cluster size is also done by calculating node distribution across clusters.

5.3.3.2 Drug Raking in Clusters

Cluster ID	Degree	Closeness Centrality	Betweenness Centrality	PageRank	Eigenvector Centrality
CW_C1	sorafenib	(-)-cubebin	sirolimus	sorafenib	sirolimus
	sirolimus	regorafenib	sorafenib	curcumin	sorafenib
	curcumin	cabozantinib	curcumin	sirolimus	curcumin
CW_C2	L-threonine	desmosine	L-threonine	L-threonine	L-threonine
	L-serine	L-seryl group	L-serine	L-serine	L-serine
	L-phenylalanine	L-tyrosine	L-phenylalanine	L-phenylalanine	L-threonine residue
CW_C3	Glu-Phe-Val	L-gamma-glutamyl-L-cysteine	ombitasvir	Glu-Phe-Val	Thr-Ser
	Phe-Asn	Glu-Met	Thr-Trp	Phe-Asn	Thr-Trp
	Thr-Ser	Ala-Asp-Pro	Thr-Ser	Thr-Ser	Glu-Phe-Val
CW_C4	tandutinib	PD-153035	tandutinib	tandutinib	tandutinib
	afatinib	linifanib	afatinib	afatinib	afatinib
	ibrutinib	ibrutinib	ibrutinib	GSK690693	ibrutinib

Table 19: Top 3 drugs in each cluster for experiment 4.

5.3.4 Second Iteration & Louvain Modularity (Resolution 1)- Experiment 5

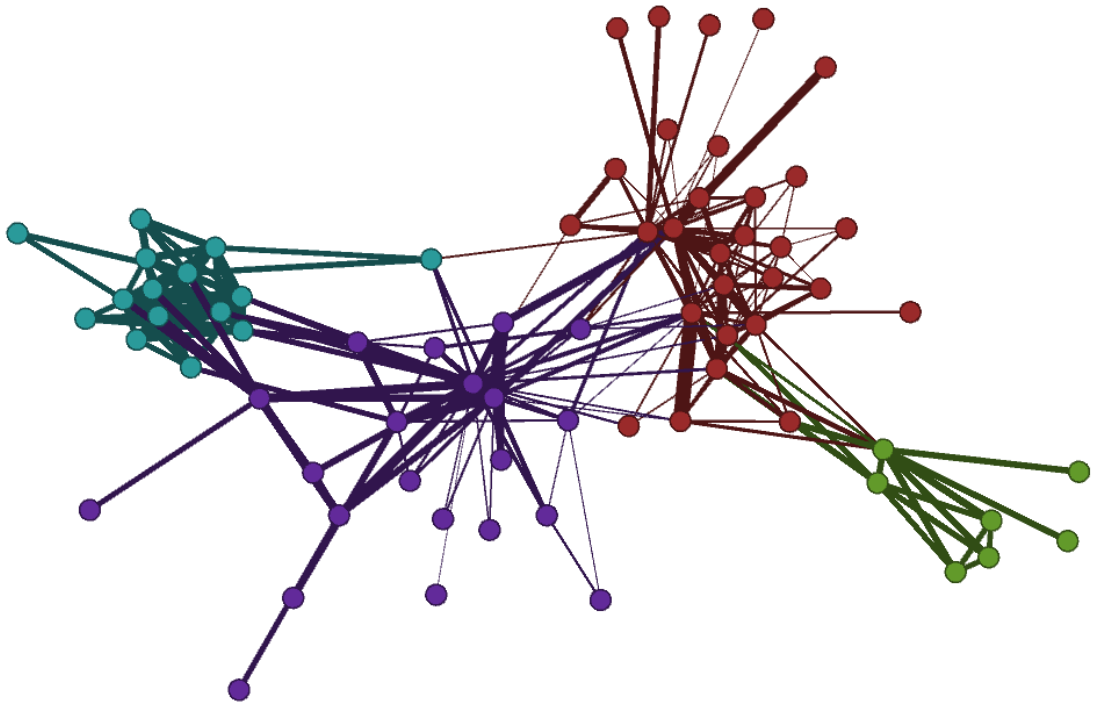


Figure 18: Yifan Hu representation for clusters in 2nd Iteration using LM[R=1] algorithm.

5.3.4.1 Cluster Distribution:

Cluster ID	Colour	Distribution
LM1_C1		40.28
LM1_C2		29.17
LM1_C3		20.83
LM1_C4		9.72

Table 20: Clusters in experiment 5.

- LM[R=1] identified four clusters as shown in figure 18 drawn using Yifan Hu graph representation algorithm.
- In Table 20 a cluster identification is assigned to each cluster, colour of each cluster is also shown corresponding to figure 18 and comparison of cluster size is also done by calculating node distribution across clusters.
- Total modularity of the graph is 0.556.

5.3.4.2 Drug Raking in Clusters

Cluster ID	Degree	Closeness Centrality	Betweenness Centrality	PageRank	Eigenvector Centrality
LM1_C1	sorafenib	(-)-cubebin	sirolimus	sorafenib	sirolimus
	sirolimus	regorafenib	sorafenib	curcumin	sorafenib
	curcumin	cabozantinib	curcumin	sirolimus	curcumin
LM1_C2	L-threonine	desmosine	L-threonine	L-threonine	L-threonine
	L-serine	L-seryl group	L-serine	L-serine	L-serine
	L-phenylalanine	L-tyrosine	L-phenylalanine	L-phenylalanine	bortezomib
LM1_C3	Glu-Phe-Val	L-gamma-glutamyl-L-cysteine	ombitasvir	Glu-Phe-Val	Thr-Ser
	Phe-Asn	Glu-Met	Thr-Trp	Phe-Asn	Thr-Trp
	Thr-Ser	Ala-Asp-Pro	Thr-Ser	Thr-Ser	Glu-Phe-Val
LM1_C4	tandutinib	PD-153035	tandutinib	tandutinib	tandutinib
	afatinib	linifanib	afatinib	afatinib	afatinib
	ibrutinib	ibrutinib	ibrutinib	GSK690693	ibrutinib

Table 21: Top 3 drugs in each cluster for experiment 5.

5.3.5 Second Iteration & Louvain Modularity (Resolution 0.6)- Experiment 6

Figure 19: Yifan representation for clusters in 2nd Iteration using LM[R=0.6] algorithm.

5.3.5.1 Cluster Distribution:

Cluster ID	Colour	Distribution
LM0.6_C1		29.17
LM0.6_C2		23.61
LM0.6_C3		19.44
LM0.6_C4		18.06
LM0.6_C5		9.72

Table 22: Clusters in experiment 6.

- LM[R=0.6] identified 5 clusters as shown in figure 19 drawn using Yifan Hu graph representation algorithm.
- In Table 22 a cluster identification is assigned to each cluster, colour of each cluster is also shown corresponding to figure 19 and comparison of cluster size is also done by calculating node distribution across clusters.
- Total modularity of the graph is 0.219.

5.3.5.2 Drug Raking in Clusters

Cluster ID	Degree	Closeness Centrality	Betweenness Centrality	PageRank	Eigenvector Centrality
LM0.6_C1	L-threonine	desmosine	L-threonine	L-threonine	L-threonine
	L-serine	L-seryl group	L-serine	L-serine	L-serine
	L-phenylalanine	L-tyrosine	L-phenylalanine	L-phenylalanine	bortezomib
LM0.6_C2	sorafenib	(-)-cubebin	sorafenib	sorafenib	sorafenib
	curcumin	regorafenib	curcumin	curcumin	curcumin
	lenvatinib	cabozantinib	ombitasvir	lenvatinib	lenvatinib
LM0.6_C3	Glu-Phe-Val	L-gamma-glutamyl-L-cysteine	Thr-Trp	Glu-Phe-Val	Thr-Ser
	Phe-Asn	Glu-Met	Thr-Ser	Phe-Asn	Thr-Trp
	Thr-Ser	Ala-Asp-Pro	Glu-Phe-Val	Thr-Ser	Glu-Phe-Val
LM0.6_C4	sirolimus	doxycycline	sirolimus	sirolimus	sirolimus
	vemurafenib	2-(2-amino-3-methoxyphenyl)chromen-4-one	selumetinib	selumetinib	selumetinib
	selumetinib	dabrafenib	trametinib	vemurafenib	vemurafenib
LM0.6_C5	tandutinib	PD-153035	tandutinib	tandutinib	tandutinib
	afatinib	linifanib	afatinib	afatinib	afatinib
	ibrutinib	ibrutinib	ibrutinib	GSK690693	ibrutinib

Table 23: Top 3 drugs in each cluster for experiment 6.

5.3.6 Analysis

Similar to first iteration, experiments across different algorithms in the second iteration produced analogous results to each other.

1. There was no super or sub-clustering relationship between LM[R=1] and CW. Both algorithms produced similar clusters with respect to node composition. Although, there

were some minor changes in border nodes between the top three clusters with the highest distribution in clusters produced by both algorithms.

2. Total number of clusters was the same for LM[R=1] and CW but LM[R=0.6] had one more cluster.
3. Both experiments, LM[R=1] and LM[R=0.6] showed an acceptable overall modularity in their clusters.

CW produces 4 clusters, LM[R=1] also produces 4 clusters with slight variation and LM[R=0.6] produces 5 clusters. Figure 20 shows the breakdown of clusters produced by three algorithms for second iteration.

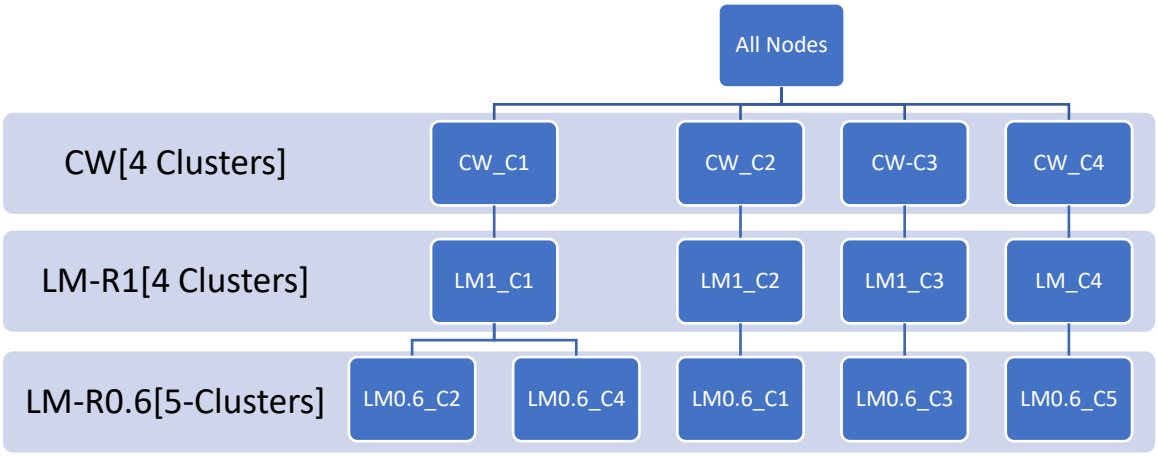


Figure 20: Relationship between cluster of different algorithms in second iteration.

While considering the ranking of top 10 drugs across centralities for the whole network, 6 similarities across 4 centrality measures of ranking were discovered. Sirolimus, sorafenib, curcumin, L-threonine and selumetinib drugs were common across Degree, Betweenness, PageRank and Eigenvector centrality.

Clusters which extend each other in figure 20 were almost alike. This signifies that clusters which were in the same branch of hierarchy shown in the figure contain almost similar nodes excluding some boundary changes. When top 3 drugs across related clusters (as shown by figure 20) were calculated it showed overlapping results.

i. Top 3 drugs across 5 parameters in CW_C4, LM_C4 and LM0.6_C5 are exactly the same.
ii. Top 3 drugs in CW_C3, LM1_C3 and LM0.6_C3 clusters have just one dissimilarity in betweenness centrality.
iii. Top 3 drugs in CW_C2, LM2_C2 and LM0.6_C1 has just one dissimilarity in eigenvector centrality.

iv. For first branch extending from CW_C1, top 3 drugs are same for CW_C1, LM1_C1 and LM0.6_C4, whereas LM0.6_0.4 is completely different.

There overall result is consistent with the fact that the clusters of a branch are very similar in size and composition. Hence, top 3 drugs in different clusters of the same branch are very similar.

5.4 Trends across Bothe Data sets

- By merely visualising the graph, it can be discerned that a drug might show more similarity to some drugs than to others i.e. a drug may be more interconnected with one cluster and might not have any connection to drugs in another cluster of the graph.
- Drug pairs which share high weight between each other are usually connected to each other by more than one path. For example: if NODE A connects to NODE B directly with a high weight, then it is highly likely that NODE A shares another connection to NODE B via neighbouring nodes with just two or more edge jumps.
- Top 10 ranked drugs for whole network were compared in both first and second iterations across 4 centrality parameters to check for similarities while discarding their ranking order. Due to the method by which closeness centrality is calculated, it is already known that comparing closeness across different iterations will not yield any overlap. Table 24 highlights similarities in first iteration and second iteration across Degree, Betweenness, PageRank and Eigenvector centrality.

S#	Degree	Betweenness Centrality	PageRank	Eigenvector Centrality
1	curcumin	curcumin	curcumin	curcumin
2	L-threonine	L-threonine	L-threonine	L-threonine
3	selumetinib	selumetinib	selumetinib	selumetinib
4	sirolimus	sirolimus	sirolimus	sirolimus
5	sorafenib	sorafenib	sorafenib	sorafenib
6	trametinib	trametinib	trametinib	trametinib
7	L-serine	L-serine	L-serine	-
8	vemurafenib	-	vemurafenib	vemurafenib
9	-	-	-	PLX-4720
10	-	-	-	lenvatinib

Table 24: Common drugs across 4 centrality parameters for first and second iteration.

1. The results signify that for top 10 drugs there is 80% overlap between 4 centrality measures of first and second iteration. This shows that most important drugs had already been identified in inner join data set and additional drugs added by outer join do not have any significant change in top 10 drugs. This indicates that the drugs which are most important were also the ones which had their relationships to other drugs verified by both molecular and gene ontologies.
2. 6 out of 10 drugs were common across 4 centralities, serial number 1 to 6 as mentioned in the table. That means these drugs hold significant importance due to their presence in top 10 drugs across 4 centrality parameters of both inner and outer join data sets. Secondly, it also means that different centrality measures are closely related with each other with slight variation in order of ranking except closeness.

5.5Evaluation Results

5.5.1 Internal Criterion

As already mentioned, mean silhouette coefficient is calculated for all six experiments to validate quality of results. The computed mean silhouette coefficient value is also used to do comparison between different clustering techniques.

An important aspect of silhouette coefficient is that it can be used as a parameter to optimize number of clusters in algorithms which do not automatically determine number of clusters in the network such as k-means. Multiple experiments are performed while varying number of clusters and calculating mean silhouette each time. The number of clusters with highest mean silhouette is chosen. This results in determining optimized silhouette coefficient value. This setup is not used for this research and silhouette coefficient is only computed to validate clusters.

In this research study, all techniques that were used to calculate clusters estimate the number of clusters automatically based on optimisation of different parameters other than silhouette coefficient. This presents an advantage that we do not need to decide on number of clusters. Though, the disadvantage is that automatically chosen number of clusters will not result in optimized silhouette parameter. Different clustering techniques rely on different parameters to optimize the number of clusters. CW relies on random class distribution with fixed number of iteration and LM relies on optimizing modularity. To test the cluster quality silhouette is

calculated for all clusters. It is a non-biased parameter as both the algorithms do not try to optimize it directly.

			Algorithms		
			Chinese Whisper (CW)	Louvain Modularity [Resolution 1] (LM[R=1])	Louvain Modularity [Resolution 0.6] (LM[R=0.6])
Iteration	First	Inner Join	0.54	0.21	0.14
	Second [After Post Processing]	Outer Join	0.050	0.042	0.024

Table 25: Result of mean silhouette coefficient against each iteration and algorithm.

The results are consistent with what was expected. First iteration results are considerably good as they are higher in positive range compared with second iteration. Performance across algorithms shows same trends across different iterations. CW algorithm performs considerably good followed by LM with Resolution 1 and LM with resolution 0.6 showed lowest silhouette in both iterations. Results suggest that the more we try to break down the network into smaller clusters, the poorer the cluster quality gets.

Some important observations are:

1. CW shows highest and lowest silhouette coefficient. Highest is for first iteration, it's a densely connected structure split into two parts. Lowest is for LM[R=0.6] in second iteration, where the network is divided into 5 clusters.
2. CW is only practical for dense structures and fails to produce satisfactory results where small world property is not met. On the other hand, LM is an easy to go algorithm, it's not very efficient but shows flexibility in terms of applicability to different structures.

There are no guarantees that one particular algorithm would produce better silhouette coefficient. It really depends on the graph structure itself which derives the decision of cluster quality for different algorithms. Nonetheless our results suggest that CW produced better silhouette coefficient.

In the second iteration, there were some broken nodes unconnected to the main structure. These unconnected nodes were removed in a post processing step. To compare the effect of

the randomized broken nodes on the overall structure, mean silhouette coefficient was calculated before removing the nodes. The results are shown in the table 26.

			Algorithms		
			Chinese Whisper (CW)	Louvain Modularity [Resolution 1] (LM[R=1])	Louvain Modularity [Resolution 0.6] (LM[R=0.6])
Iteration	Second [With All Nodes]	Outer Join	-0.39	-0.36	-0.26

Table 26: Mean silhouette coefficient in second iteration before post-processing step.

As clearly seen from the table, all the values are in negative range which suggest poor clustering performance. In comparison, value of mean silhouette is slightly low for each algorithm before and after applying post processing step. Although some broken nodes combine together to form structures with more than 2 nodes and the fact that they were separate from the larger structure also adds to their credibility. But their smaller size and no link with the larger structure suggest that drugs being considered were either extremely unique or might not be related to cancer at all. In order to ensure reliability, a post processing step to remove these smaller unconnected nodes was necessary.

5.5.2 External Criterion

The result of clustering was reviewed by an expert. As already explained in Section 4.6.2, due to time limitations clusters were not reviewed in detail. Although, some test cases of drugs that show similar MOA were tested.

Following are the domain expert's opinion/comments:

1. Some of the test cases of drugs were successfully checked out by the expert and in his opinion the research showed promising results.
2. He recommended extending the research by adding more cancer drugs from annotated drug data sets such as drug bank in-order to detect relationships of drugs from different perspectives such as highlighting cancer drugs that are closely associated to each other or highlighting specific drugs that might be strongly associated with melanoma.
3. He seemed convinced that further research into highlighting dominant MOA of each cluster may uncover important relationships about drug interaction.

4. Another advice was to use hierarchical clustering. The clustering algorithm not only clusters the nodes but also provides a tree structure in which these clusters can be combined together to form a super-cluster on different levels. This tree structure is called dendrogram and is useful when dealing with unsupervised learning in bio informatics domain as it gives the ability to combine sub clusters into super-clusters, if needed.

Clustering drugs based on molecular and gene ontologies is a new approach. Experts seemed confident about viability of the research and expressed that it holds great potential.

5.5.3 Comparison of Chinese Whispers Vs Louvain Modularity

Clusters across CW and LM produced consistent results with little degree of variation in boundary for both first and second iteration. Although the relationship between super and sub cluster can be seen at some places but the base cluster remained the same. Overall, this adds confidence in our research as the results seems to be stable even across multiple algorithms.

Chapter 6: CONCLUSION

The research revolved around the idea of highlighting similarity and dissimilarity between drugs i.e. putting groups of drugs together which show similarity with respect to drug ontologies and putting drugs into separate groups which show dissimilarity with respect to drug ontologies. In this research, 831 melanoma related cancer drugs were clustered in order to highlight similarities in their MOA. ChEBI was used to measure similarities in Molecular ontology whereas GO was used to measure similarity in Gene ontology. Pairwise similarity was calculated between each drug and all drugs in the data set. Two different data sets were formulated; one with 'common' drug pairs in both molecular and gene ontologies, another one with 'all' drugs in molecular and gene ontologies.

Three algorithms were used for graph clustering. Two of them were original algorithms Chinese Whispers and Louvain Modularity (Resolution 1), whereas the third algorithm was a modified version of Louvain Modularity (Resolution 0.6). A comparison was carried out to highlight the relationship between their clusters. Overall, the clusters were quite consistent with some super and sub cluster relationships. The consistency in the clusters across multiple algorithms added credibility to the accuracy of results.

Strict parameters were set for data processing so that only drugs with considerable high similarity were shown in the clusters. Out of 831 input drugs only 43 drugs were clustered for inner join iteration and 72 drugs were clustered for outer join iteration. This was done to ensure reliability of the results and reproducibility. Although, both algorithms were not deterministic but strict matching ensured that the results were stable to a certain confidence. Clustering algorithms were repeatedly applied and similar results were observed. Base cluster remained same but slight variation were found in cluster boundaries. As a result of the experiments 4 vital relationships were highlighted inside the data set.

1. Firstly, drugs which belong to a cluster share the same dominant MOA and secondly, drugs which are in different clusters are dissimilar or have low similarity with respect to their MOA. First iteration was composed of 43 drugs; CW identified two large clusters. Whereas, LM[R=1] identified four clusters; keeping one of the clusters same as CW and breaking down the second cluster of CW into further three parts. LM[R=0.6] divided the data set into five clusters keeping one of the clusters from CW the same and breaking

down the other cluster into further four parts. Second iteration was composed of 72 drugs; CW identified 4 clusters. Whereas, LM[R=1] identified the same four clusters as identified by CW. LM[R=0.6] on the other hand divided the data set into five clusters; producing three clusters similar to CW and breaking down one cluster of CW into further two parts.

For first iteration, CW gave the best silhouette coefficient result as the clusters are divided into just two distinct clusters. The clear distinction in clusters came at a cost of low number of clusters. This meant, as the number of cluster were raised the distinctions between clusters became less clear, the confidence of separation was reduced and the value for silhouette coefficient was decreased. None of the clusters were wrong, each set of clusters represented varying levels of distinctions that the algorithm was able to detect based on efficiency of algorithm. Though, these results further needed to be analysed by domain experts to choose the best clustering of data set.

2. Third important relationship extracted was that all drugs in first and second iteration were ranked in order of their importance using five graph centrality parameters; top ten drugs are attached in this report. Six of the top ten drugs were common across four centrality measures in both inner join data set and outer join data set.
3. Lastly, drugs were ranked inside individual clusters using network centrality measures; top three drugs for each cluster are attached in this report. A comparison of top three drugs was also done between similar clusters produced by different algorithms. Result showed that top 3 clusters were consistent across similar clusters.

Every effort was undertaken to ensure credibility, viability and reproducibility of the results. Relationship between groups of drugs was highlighted and drugs were ranked. Our research is a tiny effort in a big domain of cancer research. It is our sincere hope that somebody takes the research forward to build an actionable breakthrough in cancer research. We plan to publish our results in a journal.

6.1 Future Work

There are two directions for future work. Carryout further research using this thesis as a building block for extending the research forward or improve this research by further experiments with the data sets to recover better clusters. The experiments produced clusters that highlight relationship between drugs within the clusters based on some property of MOA.

Although, clusters were identified by unsupervised learning, no effort was undertaken to identify exact MOA property shown by the clusters. Keeping this in mind, following are the areas of future work:

1. Clusters can be further analysed to experimentally find and validate MOA represented by each cluster. In the first phase, experiments can be performed focusing on identifying strongest MOA represented by each cluster. Second phase can focus on experimentally proving that the identified MOA for each cluster is correct to a certain confidence level. This would require help of a domain experts, preferably biochemists or pharmacologists.
2. This research is restricted to a tree depth of 3 levels owing to time and computational resource constraints. Improvement that can be added to the experiment is to expand the depth of ontology matching. This will result in dissemination of weights over higher range and might result in better clusters. To enable increase of tree depth, code needs to be modified to facilitate use of multi thread functionality in python.
3. Another enhancement is to apply different clustering algorithms such as MCL [75] to cluster the data sets and compare results with this research, by calculating mutual information. Clusters from different algorithms can then be overlapped to highlight inconsistencies. These inconsistencies can be further analysed with the help of domain expert to identify and drop unreliable nodes to achieve better results.
4. Other options include to explore soft partitioning clustering algorithms which has the capacity to assign more than one class to the same node. In theory, it makes sense that a drug could belong to more than one cluster, as a drug can show more than one strong MOA. The result would represent more realistic relationships.

Further research on identifying exact MOA of cluster holds enormous potential. Even observation of clusters indicates some sort of similarity between drugs, the clear distinction between groupings of drugs is a strong indicator that drugs within these groups share some property. If this property is identified and experimentally proven, it would open door to multiple frontiers. The identified relationships could be used by researchers to better understand MOA of each drug and better administer cocktails of drugs used during chemo-therapy. Furthermore, it will also bring to light new relationships that would help researchers identify potential MOA to synthesise new drugs. The possibilities are endless.

References

[1] A. M. TURING, “I.—COMPUTING MACHINERY AND INTELLIGENCE,” *Mind*, vol. LIX, no. 236, pp. 433–460, Oct. 1950.

[2] L. Torre, F. Bray, R. Siegel, and J. Ferlay, “Global cancer statistics, 2012,” *CA a cancer J.*, 2015.

[3] D. Lu *et al.*, “A survey of new oncology drug approvals in the USA from 2010 to 2015: a focus on optimal dose and related postmarketing activities.,” *Cancer Chemother. Pharmacol.*, vol. 77, no. 3, pp. 459–76, Mar. 2016.

[4] M. Kotlyar, K. Fortney, and I. Jurisica, “Network-based characterization of drug-regulated genes, drug targets, and toxicity,” *Methods*, vol. 57, no. 4, pp. 499–507, Aug. 2012.

[5] P. Csermely, V. Ágoston, and S. Pongor, “The efficiency of multi-target drugs: The network approach might help drug design,” *Trends in Pharmacological Sciences*, vol. 26, no. 4. pp. 178–182, 2005.

[6] M. Adams and C. Urban, *Pharmacology: Connections to nursing practice*. 2015.

[7] C. A. McQueen, *Comprehensive toxicology.* Elsevier, 2010.

[8] C. For and T. H. E. Chemical, *Challenges for the Chemical Sciences in the 21St Century Heal Th and Medicine*. 2004.

[9] C. C. Chang, M. A. Slavin, and S. C.-A. Chen, “New developments and directions in the clinical application of the echinocandins,” *Arch. Toxicol.*, vol. 91, no. 4, pp. 1613–1621, Apr. 2017.

[10] H. Joensuu, “Anti-Tumour Treatment Escalating and de-escalating treatment in HER2-positive early breast cancer,” 2017.

[11] C. L. Sawyers, “Cancer: Mixing cocktails,” *Nature*, vol. 449, no. 7165, pp. 993–996, Oct. 2007.

[12] A. S. Ahmad, N. Ormiston-Smith, and P. D. Sasieni, “Trends in the lifetime risk of developing cancer in Great Britain: comparison of risk for those born from 1930 to 1960,” *Br. J. Cancer*, vol. 112, no. 5, pp. 943–947, Mar. 2015.

[13] C. C. Reyes-Aldasoro, “The proportion of cancer-related entries in PubMed has increased considerably; is cancer truly "The Emperor of All Maladies"?,” *PLoS One*, vol. 12, no. 3, p. e0173671, 2017.

[14] R. Rak, A. Rowley, W. Black, and S. Ananiadou, “Argo: an integrative, interactive, text mining-based workbench supporting curation,” *Database*, vol. 2012, no. 0, p. bas010-bas010, Mar. 2012.

[15] K. Degtyarenko *et al.*, “ChEBI: a database and ontology for chemical entities of biological interest.,” *Nucleic Acids Res.*, vol. 36, no. Database issue, pp. D344-50, Jan. 2008.

[16] T. G. O. Gene Ontology Consortium, “The Gene Ontology project in 2008.,” *Nucleic Acids Res.*, vol. 36, no. Database issue, pp. D440-4, Jan. 2008.

[17] H.-J. Li, H. Wang, and L. Chen, “Measuring robustness of community structure in complex networks,” Mar. 2015.

[18] M. E. J. Newman, “The mathematics of networks.”

[19] C. Biemann and S. Teresniak, “Disentangling from Babylonian Confusion – Unsupervised Language Identification,” pp. 773–784, 2005.

[20] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” 2008.

[21] C. Biemann, “Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems,” no. June, pp. 73–80, 2006.
[22] R. Valle Bernad, “Amani Nature Reserve - an introduction,” *Rev. clínica española*, vol. 212 Suppl, pp. 1–2, 2012.
[23] S. Fortunato and M. Barthélemy, “Resolution limit in community detection,” *Proc. Natl.*, 2007.
[24] K. Okamoto, W. Chen, and X. Li, “Ranking of closeness centrality for large-scale social networks,” *Springer*.
[25] L. Freeman, “Centrality in social networks conceptual clarification,” *Soc. Networks*, 1978.
[26] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank citation ranking: Bringing order to the web.,” 1999.
[27] V. Grolmusz, “A note on the PageRank of undirected graphs,” *Inf. Process. Lett.*, vol. 115, no. 6–8, pp. 633–634, Jun. 2015.
[28] P. Bonacich, “Factoring and weighting approaches to status scores and clique identification,” *J. Math. Sociol.*, vol. 2, no. 1, pp. 113–120, Jan. 1972.
[29] S. Borgatti, “Centrality and network flow,” *Soc. Networks*, 2005.
[30] Y. Hu, “Efficient, high-quality force-directed graph drawing,” *Math. J.*, 2005.
[31] T. Kamada and S. Kawai, “An algorithm for drawing general undirected graphs,” *Inf. Process. Lett.*, vol. 31, no. 1, pp. 7–15, Apr. 1989.
[32] T. M. J. Fruchterman and E. M. Reingold, “Graph drawing by force-directed placement,” *Softw. Pract. Exp.*, vol. 21, no. 11, pp. 1129–1164, Nov. 1991.
[33] D. Khokhar, *Gephi cookbook : over 90 hands-on recipes to master the art of network analysis and visualization with Gephi*. .
[34] A. Gemma *et al.*, “Anticancer drug clustering in lung cancer based on gene expression profiles and sensitivity database.,” *BMC Cancer*, vol. 6, p. 174, Jun. 2006.
[35] K. Uhr *et al.*, “Understanding drugs in breast cancer through drug sensitivity screening.,” *Springerplus*, vol. 4, p. 611, 2015.
[36] L. Udrescu *et al.*, “Clustering drug-drug interaction networks with energy model layouts: community analysis and drug repurposing,” *Nat. Publ. Gr.*, 2016.
[37] D. Woolner and N. Holford, “CLASS EFFECTS AND THE RATIONAL COMPARISON OF DRUGS.”
[38] D. Emig *et al.*, “Drug Target Prediction and Repositioning Using an Integrated Network-Based Approach,” *PLoS One*, vol. 8, no. 4, p. e60618, Apr. 2013.
[39] A. Brazma and J. Vilo, “Gene expression data analysis.,” *FEBS Lett.*, vol. 480, no. 1, pp. 17–24, Aug. 2000.
[40] P. Mutowo *et al.*, “A drug target slim: using gene ontology and gene ontology annotations to navigate protein-ligand target space in ChEMBL.,” *J. Biomed. Semantics*, vol. 7, no. 1, p. 59, Sep. 2016.
[41] A. Gaulton *et al.*, “ChEMBL: a large-scale bioactivity database for drug discovery,” *Nucleic Acids Res.*, vol. 40, no. Database issue, p. D1100, 2012.
[42] R. P. Huntley, D. Binns, E. Dimmer, D. Barrell, C. O’Donovan, and R. Apweiler, “QuickGO: a user tutorial for the web-based Gene Ontology browser,” *Database*, vol. 2009, no. 0, p. bap010-bap010, Sep. 2009.
[43] D. Jiang, C. Tang, and A. Zhang, “Cluster analysis for gene expression data: A survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11. pp. 1370–1386, Nov-2004.

[44] D. T. Ross *et al.*, “Systematic variation in gene expression patterns in human cancer cell lines,” *Nat. Genet.*, vol. 24, no. 3, pp. 227–235, Mar. 2000.
[45] J. Jeon *et al.*, “Network Clustering Revealed the Systemic Alterations of Mitochondrial Protein Expression,” *PLoS Comput. Biol.*, vol. 7, no. 6, p. e1002093, Jun. 2011.
[46] S.-B. Cho and H.-H. Won, “Machine learning in DNA microarray analysis for cancer classification,” *Proc. First Asia-Pacific Bioinforma. Conf. Bioinforma. 2003-Volume 19*, pp. 189–198, 2003.
[47] K. Khrabrov, P. Mamoshina, Q. Vanhaelen, and A. Aliper, “The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology,” 2017.
[48] S. Dorogovtsev and J. Mendes, *Evolution of networks: From biological nets to the Internet and WWW*. 2013.
[49] S. Elisa Schaeffer, “Graph clustering,” vol. 1, no. 2, pp. 7–6, 2007.
[50] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 99, no. 12, pp. 7821–6, Jun. 2002.
[51] M. E. J. Newman, “Modularity and community structure in networks.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, no. 23, pp. 8577–82, Jun. 2006.
[52] X. Wang, G. Chen, and H. Lu, “A very fast algorithm for detecting community structures in complex networks,” *Phys. A Stat. Mech. its Appl.*, vol. 384, no. 2, pp. 667–674, Oct. 2007.
[53] A. Clauset, “Finding local community structure in networks,” Mar. 2005.
[54] J. M. Pujol, V. Erramilli, and P. Rodriguez, “Divide and Conquer: Partitioning Online Social Networks,” May 2009.
[55] G. Karypis and V. Kumar, “A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs,” *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, Jan. 1998.
[56] L. Zhang, X. Liu, F. Janssens, L. Liang, and W. Glänzel, “Subject clustering analysis based on ISI category classification,” *J. Informetr.*, vol. 4, no. 2, pp. 185–193, Apr. 2010.
[57] T. Raeder and N. V. Chawla, “Market basket analysis with networks,” *Soc. Netw. Anal. Min.*, vol. 1, no. 2, pp. 97–113, Apr. 2011.
[58] G. Roma and P. Herrera, “Community Structure in Audio Clip Sharing,” in *2010 International Conference on Intelligent Networking and Collaborative Systems*, 2010, pp. 200–205.
[59] J. Haynes and I. Perisic, “Mapping search relevance to social networks,” in *Proceedings of the 3rd Workshop on Social Network Mining and Analysis - SNA-KDD ’09*, 2009, pp. 1–7.
[60] P. Hui and N. Sastry, “Real World Routing Using Virtual World Information,” in *2009 International Conference on Computational Science and Engineering*, 2009, pp. 1103–1108.
[61] N. A. W. van Riel, “Dynamic modelling and analysis of biochemical networks: mechanism-based models and model-based experiments,” *Brief. Bioinform.*, vol. 7, no. 4, pp. 364–374, Dec. 2006.
[62] N. A. W. van Riel and E. D. Sontag, “Parameter estimation in models combining signal transduction and metabolic pathways: the dependent input approach.,” *Syst. Biol. (Stevenage).*, vol. 153, no. 4, pp. 263–74, Jul. 2006.
[63] N. Bragazzi and C. Nicolini, “A leader genes approach-based tool for molecular genomics: From gene-ranking to gene-network systems biology and biotargets predictions,” *J Comput Sci Syst Biol*, 2013.

[64] C. von Mering *et al.*, “STRING: known and predicted protein-protein associations, integrated and transferred across organisms,” *Nucleic Acids Res.*, vol. 33, no. Database issue, pp. D433–D437, Dec. 2004.
[65] Y. Liu and A. Foroushani, “PFC: An Efficient Soft Graph Clustering Method for PPI Networks Based on Purifying and Filtering the Coupling Matrix,” *Int. Conf. Intell. Comput.*, 2016.
[66] B. Adamcsek, G. Palla, I. J. Farkas, I. Derényi, and T. Vicsek, “CFinder: locating cliques and overlapping modules in biological networks,” *Bioinformatics*, vol. 22, no. 8, pp. 1021–1023, Apr. 2006.
[67] J. Pinney and D. Westhead, “Betweenness-based decomposition methods for social and biological networks,” *Interdiscip. Stat.*, 2006.
[68] S. Ananiadou, P. Thompson, R. Nawaz, J. McNaught, and D. B. Kell, “Event-based text mining for biology and functional genomics,” *Brief. Funct. Genomics*, vol. 14, no. 3, pp. 213–230, May 2015.
[69] M. Bastian, S. Heymann, and M. Jacomy, “Gephi: an open source software for exploring and manipulating networks.,” *Icwsm*, 2009.
[70] V. R. Patel and R. G. Mehta, “Impact of Outlier Removal and Normalization Approach in Modified k-Means Clustering Algorithm,” *IJCSI Int. J. Comput. Sci. Issues ISSN*, vol. 8, no. 2, pp. 1694–814, 2011.
[71] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008.
[72] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.
[73] J. Hernández-orallo, “Introduction to Data Mining,” no. September, pp. 1–28, 2005.
[74] T. F. Coleman and J. J. Moré, “Estimation of Sparse Jacobian Matrices and Graph Coloring Blems,” *SIAM J. Numer. Anal.*, vol. 20, no. 1, pp. 187–209, Feb. 1983.
[75] A. J. Enright, D. S. Van, and C. A. Ouzounis, “An efficient algorithm for large-scale detection of protein families,” *Nucleic Acids Res.*, vol. 30, no. 7, pp. 1575–1584, 2002.