DISEASE SIMILARITY USING BIOLOGICAL MODULE DYSREGULATION PROFILE

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Eshita Zaman

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

October 2016

Fargo, North Dakota

NORTH DAKOTA STATE UNIVERSITY

Graduate School

Title

DISEASE SIMILARITY USING BIOLOGICAL MODULE DYSREGULATION PROFILE

By

Eshita Zaman

The supervisory committee certifies that this paper complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Saeed Salem
Chair

Dr. Simone Ludwig

Dr. Mukhlesur Rahman

Approved:

November 18, 2016
Date

Brian M. Slator
Department Chair

ABSTRACT

Diseases can be grouped according to phenotypic and genotypic similarities. Gene expression and micro-RNA data paved the way to look inside the genetic coding and classify diseases accurately. Modern system biology seeks to understand the underlying protein complexes in a cell and how they are altered in disease condition. In this research, we aimed to mine cohesive biological modules from large micro-RNA dataset and show the genes in these modules are dysregulated in a number of diseases. We used 13 different types of cancer and DME algorithm to extract dense modules satisfying a user defined density. Binary attribute profiles of genes are also provided. We have shown that disease similarity based on the average module dysregulation yield disease pairs that share common disease genes. Collectively, we have concluded that the recurrence of these modules in different cancer types increase the therapeutic opportunity to treat more diseases with existing drugs.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my adviser, Dr. Saeed Salem. Only by his leadership was I able to produce this work. His ambition and passion for the subject inspired my motivation.

I extend special thanks to my family for encouraging me to work hard, pursue further education and a graduate degree.

I would also like to thank my committee members, Dr. Simone Ludwig and Dr. Mukhlesur Rahman, for taking the time to evaluate my project and helping me to graduate from the University.

I also want to express my thanks for North Dakota State University and the Department of Computer Science for offering me the opportunity to study at such a great school. The people and the environment have made it a great place to study and learn.

Finally, I would like to thank my fellow research mate, Bassam Qormosh for working with and beside me during my research and helped me in many ways to improve my findings.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

In recent years, lot of research have been done to find the similarity between diseases. With the advancement of technology and micro-array chips, researchers have established disease-disease relationship using genomic data (Suthram et al. [15]). It has been proved that the activity of a gene variant in a complex process depends on the presence of another gene variant in that process. It is called the epistatic effect. Singular gene variant has very small impact on a specific cellular functionality. Additive approach of calculating each gene effect and finally summing them up does not represent the whole impact of dysregulated (having values above or below normal values) genes in a complex disease trait [12]. It guides scientists' to look for dysregulated gene modules responsible for disease state. In order to find disease correlation, we tried to find modules of genes that are dysregulated in disease sample and also interact in actual protein-protein interaction (PPI) network. In addition to genetic interaction, it is often important to consider attribute profiles of genes which can give more contextual information. By combining information about the relational structure of the network and the properties of each gene in it, we are able to extract more meaningful results.

Due to the large amount of data involved in biological network analysis, researchers have employed data mining techniques to filter out meaningful and relevant portions. The Biological General Repository for Interaction Datasets (BioGRID) [1] is a database that contain manually curated scientific information pertaining to the biology of most human proteins and genetic interactions. Information regarding proteins involved in human diseases is annotated and linked to Online Mendelian Inheritance in Man (OMIM) database. The National Center for Biotechnology Information provides link to these databases through its human protein databases (e.g. Entrez Gene, RefSeq protein) related to genes and proteins. All the protein interaction databases are updated regularly.

To conduct our research, we used the concept of itemset mining from data mining techniques. Frequent itemset mining is an interesting branch of data mining that focuses on looking at sets of actions or events. In frequent itemset mining, the base data takes the form of sets of instances (also called transactions) that each has a number of features (also called items). The task for the frequent itemset mining algorithm is then to find all common sets of items, defined as those itemsets

that have at least a minimum support (exists at least a minimum amount of times). Biological modules discovery is highly useful in biological network analysis for finding patterns that lead to particular behaviors, such as a meaningful protein complex.

For functional prediction of genes and proteins, extracting dense interacting genes modules is undoubtedly useful. The search for correlated gene expression patterns is usually achieved by clustering them. Density based graph clustering algorithms are used to detect network modules [20]. It uses the local density of genes to determine the clusters, rather than using only the distance between genes. In this experiment, we used DME (Dense Module Enumerator) algorithm [6] to extract dense modules of genes from PPI network. DME presents a method to enumerate all modules that exceed a given density threshold. It also integrates more constraint like binary attribute profile of genes. We used DME to report dense module of genes with similar profiles which are frequent in experimental dataset.

For the remainder of this paper, we begin by reviewing the related works of finding groups of genes responsible for diseases in section 2. Section 3 provides some necessary definitions to understand the methods and algorithms we used. In Section 4 we introduce our experiments and explain its operation with examples. In Section 5 we discuss and analyze the results of our experiments. Finally, Section 6 outlines the conclusion and sheds light to the future work of our experiments.

2. RELATED WORK

Traditionally, a network is often represented as a graph where a person in a social network or a protein in a PPI network can be represented as a vertex and an interaction between them is denoted by an edge. The attribute profiles of entities are often represented as a function mapping of vertices to vectors of real number values or binary values. The idea of extracting meaningful information from a large amount of data is not new. It has been done in different forms and in different aspects of life. There are a number of existing approaches that use graph mining techniques to discover meaningful patterns in large (and possibly attributed) networks [13]. We will discuss few of them in the following paragraphs.

2.1. Text Mining

Text mining, also known as text data mining, is the process of discovering useful information from unstructured text. Large amount of unstructured text are produced everyday in many application scenarios. In text mining, the goal is to discover unknown information from natural language text document. It focuses on the process of extracting text data of some kind of relevance and interestingness. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning [16]. Text mining tasks include classification, clustering of text document, entity extraction,terminology extraction, relationship extraction and hypothesis generation [4]. A regular example of text mining is the categorization of online news and opinion mining. Online news portals scan the news and put them together in a group depending on the word content of the articles. If the world 'flood' appears most, that means the news is in natural calamity category.

2.2. Data Mining

Data mining is a buzzword and is frequently applied to any form of large-scale data processing. It is the computational process of discovering hidden knowledge from huge amount of raw or unprocessed data. It is no longer the research area of computer science rather it's frequently used in many aspects of life. Data mining incorporate techniques from the fields of artificial intelligence, machine learning, statistics, and database systems [5]. The overall goal of data mining is to extract meaningful information from data and transform it into an understandable form for

further use. Apart from the analysis step, it involves data management, data pre-processing, model building for classifying data. It is also used for anomaly detection in data. Data dependencies are another aspect where data mining helps to find the underlying relationship between elements in a system [3] . Briefly, we can say that data mining helps us to group relevant data together and build model databases that can be used later on as a reference. Two main data mining applications are classification and clustering of data.

2.2.1. Classification

Classification is a data mining technique that assigns a new data record to one of several predefined categories or classes. It is known as supervised learning as the classes are labeled earlier. In classification, a model is trained to further classify the newly found data in a group. Mostly used classification algorithms are Decision Trees, Bayesian Networks and k-Nearest Neighbor classifiers. Typical applications of classification are credit card fraud detection, direct marketing, classifying diseases, web-pages etc.

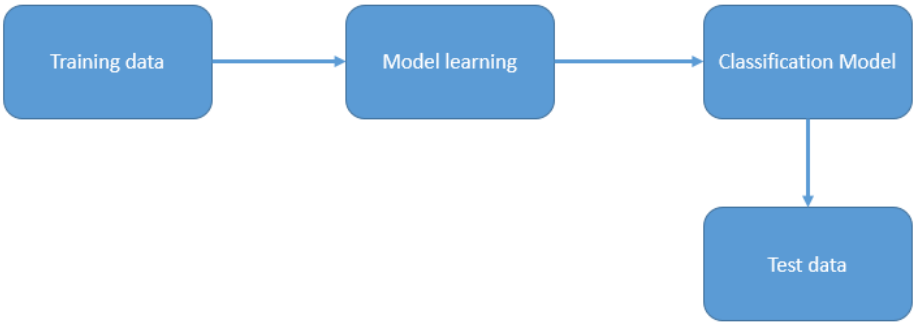


Figure 2.1. Data classification model

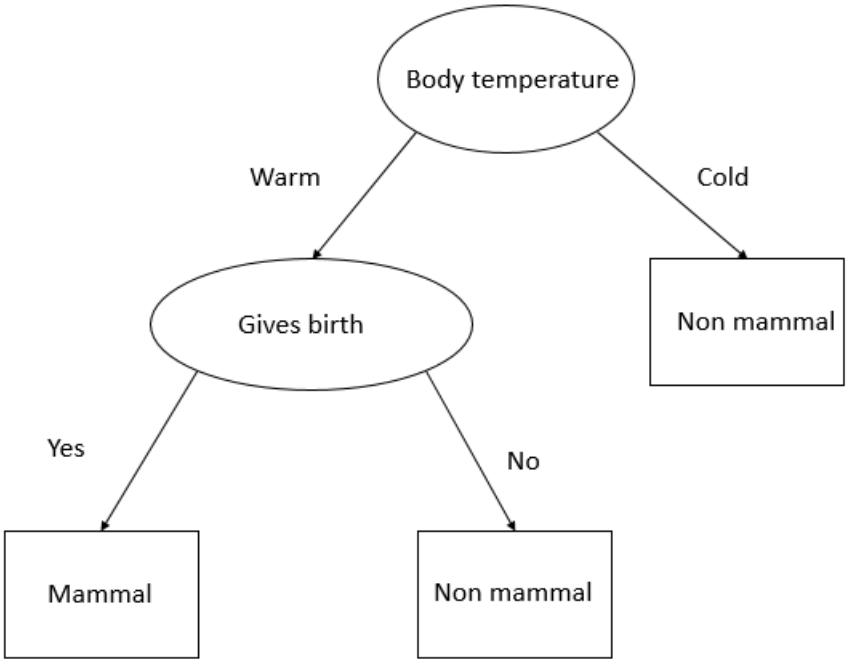


Figure 2.2. A decision tree for mammal classification problem.

Figure 2.2 shows the example of decision tree classifier. Here, mammal and non-mammal are the class labels we want to put the animals in.

2.2.2. Clustering

Clustering is data mining process that partition the dataset into subsets or groups where elements of a group share a common set of properties. It ensures that the elements of same cluster are with high intra-group similarity and small inter-group similarity. k-Means clustering [10] method is the widely used partitioning clustering techniques. Other than that Hierarchical or density based clustering are also done. In clustering the number of clusters or groups can be predefined or not. It is called unsupervised learning as there is no predefined classes to put the newly discovered data into it.

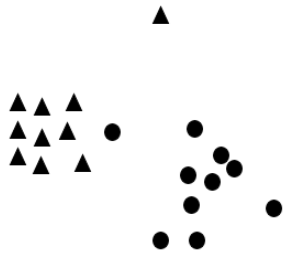


Figure 2.3. Original points

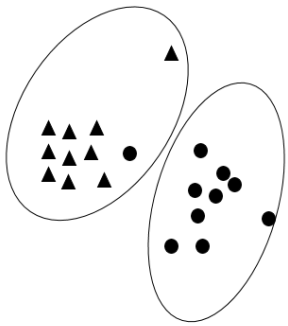


Figure 2.4. Two cluster of points

Figure 2.4 shows the partition of points into two clusters with maximum similarity of points in one group.

2.2.3. Pattern Mining

Pattern mining is concerned with finding statistically relevant patterns (a set of items, subsequences, substructures, etc.) that occur in a set of transactions. It is mainly known as sequencial pattern mining. A pattern that occurs frequently in a data set is called frequent pattern. The idea of finding frequent pattern was first proposed by Srikant and Agrawal [14]. Motivation of mining frequent pattern was to find out the underlying relationship in data like which items are bought together in superstore or what kinds of DNA sequences are sensitive to a new drug.

2.2.4. Itemset Mining

Nowadays the quest to mine frequent patterns appears in many other domains. The most common application is to mine the sets of items that are frequently bought together at a super-

market. This is called market basket analysis. Once we mine the frequent sets, they allow us to extract association rules among the item sets, where we make some statement about how likely are two sets of items to co-occur or to conditionally occur.

Let $I = \{x_1, x_2, ..., x_m\}$ be a set of elements called items. A set $X \subset I$ is called an itemset. An itemset of size k is called a k-itemset. Let $T = \{t_1, t_2, ..., t_n\}$ be another set of transaction identifiers or tids. A transaction is a tuple of the form (t, X_i), where $t \in T$ is a unique tid, and X is an itemset [20]. A set of transaction is called the transaction database. Support of items in X is calculated by number of occurrence of that item in the transaction. It's represented as sup(X,D) where D represents the database [20]. From Table 2.1 we get $t(A) = 1, 2, 3$ and $t(\{A, C\}) = 1$. Support of A and A, C are $sup(A) = |t(A)| = 3$ and $sup(\{A, C\}) = |t(\{A, C\})| = 1$ respectively.

Table 2.1. Transaction database

Transaction id	Item
1	A,B,D
2	A,C,D
3	A,D,E
4	B,E,F
5	B,C,D,E,F

2.3. Frequent Itemset Mining

An itemset X is frequent if it satisfies $sup(X, D) \geq \delta$, where δ is a user defined minimum threshold. Testing every combination of items in a database is one way to enumerate all the itemsets [17]. After calculating their support individually, frequent itemsets can be discovered. There are several algorithms like Apriori and Eclat to mine frequent items ([21, 19]). Apriori employs a level-wise or breadth-first exploration of the itemset search space, and prunes all supersets of any infrequent candidate, as no superset of an infrequent itemset can be frequent. This property is known as **Anti-Monotone Property**. Also, it avoids generating any candidate that has an infrequent subset. Except going level-wise, Eclat improves the search time and eliminate candidates by following a DFS (depth first search) approach. Eclat intersects the tidsets only if the frequent itemsets share a common prefix and expand downward.

Transaction id	Item
1	A,B,D
2	A,C,D
3	A,D,E
4	B,E,F
5	B,C,D,E,F

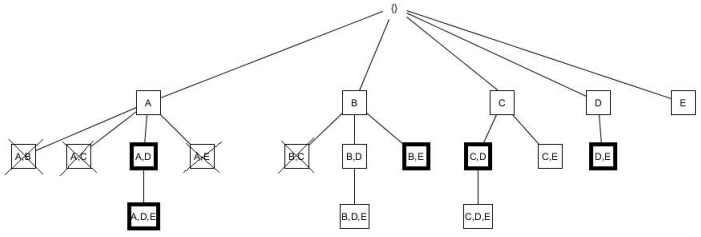


Figure 2.5. Frequent itemset enumeration tree with minimum support of 2.

Srikant and Agrawal [14], show how to use the set enumeration tree for finding *frequent itemsets* in association rule mining. With the search space growing vastly with each increase in set size, there is a need to prune the search space to avoid taking too much time. In the itemset enumeration tree, *anti-monotone property* allows to prune all child node if the parent node is not frequent. Given the example dataset in Table 2.1, let minsup = 2. Itemset BE is contained in tids

4 and 5, so $t(BE) = 45$ and $sup(BE) = |t(BE)| = 2$. Thus, BE is a frequent itemset. But itemset AC appeares in only transaction 2 and $sup(AC) = |t(AC)| = 1$. So, AC is not frequent itemset. In Figure 2.5, the search branches rooted at {A,B}, {A,C}, {A,E}, and {B,C} are pruned this way. The frequent itemsets are shown as bolded boxes in Figure 2.5.

2.4. Networks

With the invention of internet the whole world is now connected. People belong to the same institution, living in the same area, and/or having the same religion are connected. Again, through social networks, people all over the world are connected. Even the chemical compounds can be modeled as networks, all the protein reactions in cell can be represented as networks. Analyzing the networks, we get to know the pathways how these communications take place.

(a) Network of 8 nodes

(b) Subnetwork of 2.6a

Figure 2.6. Example of networks

2.5. Graph Mining

Graphs have become increasingly important in modeling complicated structures, such as circuits, images, biological networks, social networks, the Web, and XML documents. Many graph search algorithms have been developed in chemical informatics, computer vision, video indexing, and text retrieval.

Graph means an interconnected set of entities. If we look deeper, we find some common properties of those entities that bind them together and form a group. Examining those properties, one can get an overall idea of the group. Analysis of a single graph is the building blocks for graph classification, clustering, compression, comparison, and correlation analysis of large networks. With

the increasing demand on the analysis of large amounts of structured data, graph mining has become an active and important theme in data mining [11]. But graph mining is costly. Several algorithms have been developed to make graph mining efficient and result set useful for real life problems. In this experiment, we worked with one of them named DME algorithms [6].

2.5.1. Enumerating Dense Subgraphs

To generate the set of interactive gene modules, we need to consider the density as well. If we get a set of genes that are loosely connected, it will not satisfy our goal. But very few algorithms consider density to generate candidates. In 2007, Uno et al. [18] introduced an algorithm for mining dense modules using the traditional *density* definition. In a θ-dense subgraph, the ratio of the number of edges to the total possible number of edges is at least $0 \leq \theta \leq 1$, where θ is a user-defined minimum threshold. This definition allows for more flexibility because it observes the density of the pattern as a whole rather than concerning each individual member. Again, we want to pass the attribute profile as an important criteria to generate candidate subgraphs.

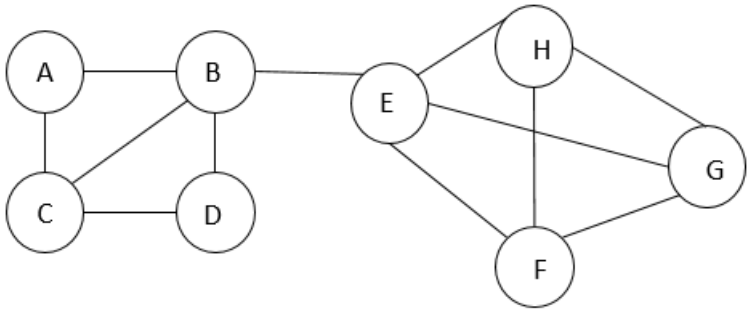


Figure 2.7. An example of dense subgraph

The density of a subgraph U is defined as

$$\rho(U) = \frac{|E(U)|}{\frac{(|U|-1)*|U|}{2}}$$

E(U) is the edgeset and $|U|$ is the cardinality of subgraph U. In Figure 2.7, the density of subgraph containing vertices A,B,C,D is $\rho(\{A, B, C, D\}) = \frac{5}{6}$.

Uno et al. [18] presented a method for efficiently enumerating dense subgraphs and later contributed development of the Dense Module Enumeration (DME) algorithm [6]. DME mines maximal dense subgraphs (or *modules*) from weighted networks. They provide the definition for the weighted degree of a vertex as the sum of its edge weights. For example, in Figure 2.9, the weighted degree of C = 0.43. In addition, they define a *module density* equation $\rho(U)$ which amounts to the sum of the edge weights in the module U divided by the number of possible edges in the module. Formally,

$$\rho(U) = \frac{\sum_{i,j \in U, i<j} w_{ij}}{|U|(|U|-1)/2}$$

where w_{ij} is the weight of the edge between i and j. The algorithm takes advantage of a property for these dense graphs which states that the density of a module does not increase when a vertex added to the module has weighted degree that is no larger than the weighted degree of each other vertex already in the module; i.e. if $v \in U$ is a node with minimum wighted degree in U: $\forall u \in U : deg_U(u) \geq deg_U(v)$. Then, $\rho(U \backslash \{v\}) \geq \rho(U)$. Inversely, it is also true that removing a vertex with minimum weighted degree in U does not decrease the density of U. They provide a proof for this property in [18, 6] and explain how they are able to leverage it in order to prune unnecessary branches in the enumeration tree. With this knowledge, patterns can be discovered in such a way that, as the enumeration tree is expanded from top to bottom, module sizes are increasing while their densities are guaranteed to be decreasing or remaining the same. Therefore, if a module with density less than a given threshold θ is found in the tree, we can stop extending it since none of its children can pass the threshold. Before running the algorithm, the vertices in the graph must be given a strict order. The example in Figure 2.9 uses an ordering of $ord(A) < ord(B) < ord(C) < ord(D) < ord(E)$. Then, the procedure begins with the empty set and builds the enumeration tree. Let U be the set of vertices at the current node in the tree and $Z = V \backslash U$ be the remaining vertices in the graph that are not in U. At each stage in the enumeration, a branch of the tree is extended with $z \in Z$ to produce $U' = U \cup \{z\}$ if one of the following conditions are met:

- The weighted degree of z w.r.t. U' is strictly less than each other vertex in U.
- The weighted degree of z w.r.t. U' is equal to the weighted degree of each other vertex in U **and** the order of z is less than the order of each other vertex in U.

More formally,

$$\forall u \in U : (deg_{U'}(z) < deg_{U'}(u)) \vee (deg_{U'}(z) = deg_{U'}(u) \wedge ord(z) < ord(u))$$

DME traditionally mines maximal patterns, which are circled in the image. The pruned branches are crossed out.

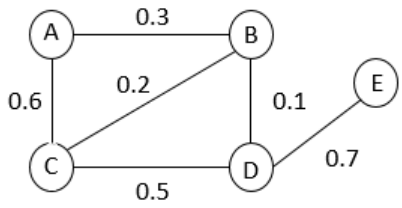


Figure 2.8. Weighted graph

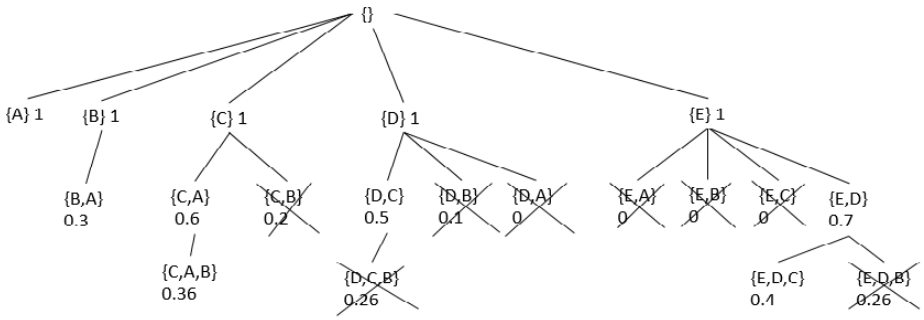


Figure 2.9. DME enumeration tree of graph given in Figure 2.8

For this tree, $\theta = 0.3$ and the order of vertices are lexicographical. Crosses show which branches are able to be pruned.

The method only outputs locally maximal solutions, i.e. modules where all direct super-modules (containing one additional node) do not satisfy the minimum density threshold. DME employs a reverse search strategy, which allows to exploit the density criterion in an efficient way and it is proved that it finds biologically meaningful modules.DME additionally includes limited subspace clustering of the attribute profiles of vertices.The ability to process real data is an important feature for many network mining algorithms.

3. PROBLEM DEFINITION

In this section, we will give some necessary definitions which are used to better understand the problem. As our experiment is all about finding group of genes responsible for diseases, we need to know the formal definition of disease.

Disease is the opposite of being healthy. It indicates something that is causing troubles or hampering the regular functionality of body. Reasons for disease are many, it can be caused due to genetic disorder, poison or unfavorable environmental factors. In disease condition, cell starts to produce proteins that are not necessary, sometimes harmful for body or can't control a particular process. For example, in Cancer, cell division process is uncontrolled.

Gene expression is the process by which genetic instructions are used to synthesize gene products. Gene coding are used to produce different proteins like enzymes, hormones that take part in major processes in cells.

To understand the underlying cause of diseases, scientists collect gene expression data (microarray datasets) of disease infected and healthy samples. Gene expression is the visualization of the genetic coding of DNA that gives rise to the phenotypic characteristics of that particular coding. The ratio of theses expression values ('infected' condition vs. 'healthy' condition) is called the 'fold-change' (FC). The logarithmic value of FC is called the log fold-change, abbreviated logFC. Suppose, the average value of gene A in disease samples and healthy samples is 6 and 3 respectively. So, FC value of gene A is 2, that means gene A expressed twice its actual value in disease conditions. LogFC tells us how the gene is differentially expressed in disease condition. Formally,

$$logFC = log(\frac{Avg^{+}(gi)}{Avg^{-}(gi)})$$

'gi' represent any particular gene in the expression data. We are only interested in gene's which are dysredulated in disease samples.

The best way to represent network data is using graph. For example, in a protein-protein interaction(PPI) network, a protein is represented by a vertex and the interaction between various proteins are represented by edges between the vertices. There are a number of existing approaches

that use graph mining techniques to discover meaningful patterns in large (and possibly attributed) networks. The attribute profiles of entities(vertices) are often represented as a function mapping of vertices to vectors of real number values. A graph G(V,E), where V is the set of vertices, E is the set of edges and R represents the attribute vector.

We define the *density* property (denoted as ρ) of a subgraph U similarly to the definition provided by Uno et al. [18]:

$$\rho(U) = \frac{2|E(U)|}{|U|(|U|-1)}$$

In other words, the density of a subgraph U is equal to the number of its edges divided by the number of total possible edges in U. In the case of Figure 2.7, $\rho(\{\text{C, D, E}\}) = \frac{1}{3} = 0.33$. The density of a single vertex is always 1. Intuitively, a subgraph with many edges will have a higher density value than a subset with fewer edges for the same set of vertices.

A dense subgraph $U \subseteq V$ is an induced subgraph of G, given a density threshold parameter $0 < \theta \leq 1$, the subgraph U has a density value $\rho(U) \geq \theta$. Suppose, $\theta = 0.8$. For the graph in Figure 4.1, $\rho(U\{A, B, C\}) = 1$, $\rho(U\{A, B, C, D\}) = .83$. So, the resultant dense subgraph is U{A,B,C,D}.

Additionally, we need to introduce the notion of *support* . Support is an indication of how frequently the item-set appears in the dataset. The support of an itemset X in a dataset D, denoted sup(X,D), is the number of transactions in D that contain X: $sup(X, D) = \|\{t \| \langle t, i(t) \rangle \in \text{D} and \text{X} \subseteq i(t)\}\| = \|t(X)\|$

A constraint P is anti-monotone for an itemset, $V \subseteq \mathcal{V}$, if the following condition is satisfied:

$$P(V) = TRUE \implies P(V') = TRUE \, , \forall \, V' \subseteq V$$

We can see that the frequency constraint is anti-monotone and that is why we can employ it in pruning search branches. Identifying small groups of related members are not very useful; instead the largest groups that still satisfy the frequent property are more interesting. Thus the concept of *maximal* frequent itemsets is introduced. An itemset is maximally frequent if there exists no superset of that itemset which is frequent as well. Figure 3.1 shows the maximal itemsets in bold boxes.

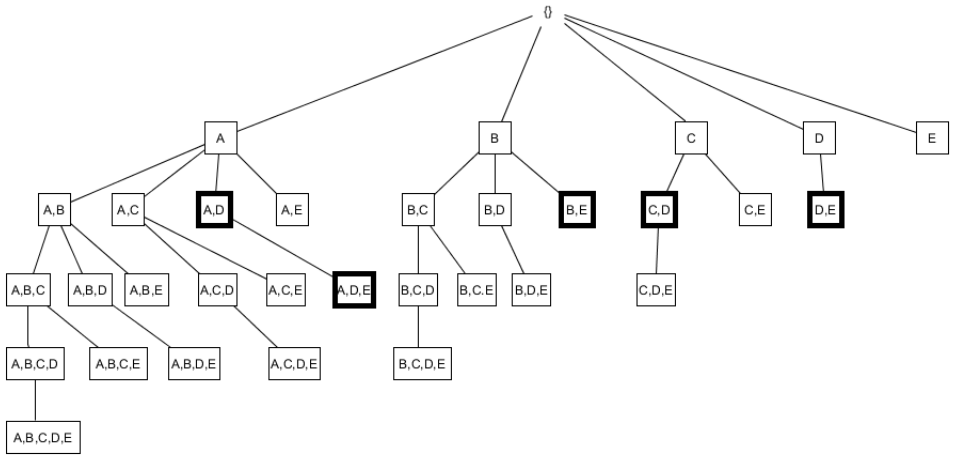


Figure 3.1. Maximal frequent itemset enumeration tree with minimum support of 2.

An itemset, $S \subseteq \mathcal{S}$, is maximal if the following condition is satisfied:

$$FREQ(S) = TRUE \, , \nexists \, S' \supseteq S \wedge FREQ(S') = TRUE$$

Enumerating only maximal itemsets offers a few more opportunities for pruning. After pruning for support value, only the leaf nodes are potential maximal frequent nodes. Also, if a node's child node is a subset of a discovered maximal set, then the node and its children can be pruned.

Even though all these constraints prune the number of itemsets from consideration, still we have a lot of them as the PPI network is huge. That's why we tried to use techniques that will find only the interesting patterns from a large attributed dataset (graph) [13].

4. MATERIAL AND METHOD

In this section, we introduce our method of getting **Co**hesive **De**nse **M**odules of genes. As the name suggests, our aim is to discover cohesive dense modules in a network – that is, dense patterns having similar attribute profiles. Gene and miRNA expression data used in this experiment in matched Cancer and normal samples were obtained from TCGA (The Cancer Genome Atlas) project (as of September 2014) by Jiang et al. [8]. To eliminate the bias from different platforms, we only considered gene and miRNA expression levels that were measured by Illunima HiSeq platform. As a result, we obtained the gene and miRNA expression data of 13 cancer types and matched normal samples; the sample sizes ranged from 14 to 172. In this study, we analyzed expression files of 13 different cancer types, including bladder urothelial carcinoma (BLCA), breast invasive carcinoma (BRCA), head and neck squamous cell carcinoma (HNSC), kidney chromophobe (KICH), kidney renal clear cell carcinoma (KIRC), kidney renal papillary cell carcinoma (KIRP), liver hepatocellular carcinoma (LIHC), lung adenocarcinoma (LUAD), lung squamous cell carcinoma (LUSC), prostate adenocarcinoma (PRAD), stomach adenocarcinoma (STAD), thyroid carcinoma (THCA) and uterine corpus endometrial carcinoma (UCEC).

We used DME algorithm to find dense module that satisfy user defined density and consistency constraints. The algorithm performs a reverse search method to discover *closed* and *maximal* dense coherent patterns with binary attribute values.

DME is originally designed for weighted network. PPI network is unweighted and we consider all the interaction weight equal and the value is '1'(if no edge is given, the interaction weight is zero by default). In Figure4.1 there is an edge between node A and B, so the interaction weight is 1. However, there is no edge between node A and D, that is why the weight is '0'. Table 4.1 represents the symmetric matrix of graph given in Figure 4.1. Let U be a subset of nodes. Then the density of U with respect to W is defined using the formula given 3. If it satisfies user defined density, we refer to these node subsets as dense modules.

In addition, the module search can respect consistency constraints with respect to external profile data. If some attribute profile for each node is available, a module is called consistent if there exists a sub-profile which is shared by all member nodes. For example, if each node of the

Table 4.1. Adjacency matrix of the graph 4.1

	A	B	C	D	E
A	0	1	1	0	0
B	1	0	1	1	0
C	1	1	0	1	0
D	0	1	1	0	1
E	0	0	0	1	0

graph corresponds to a protein, the profile could indicate genes' presence or absence across multiple cellular conditions. For each of the states (for example, presence or absence), the user can define the minimum required number of profile conditions for which all module members are in the same state. So, using DME we can systematically mine for dense modules with interesting profiles. Figure 4.2 shows the resultant modules of graph 4.1 with density 0.3 and minimum attribute length 3.

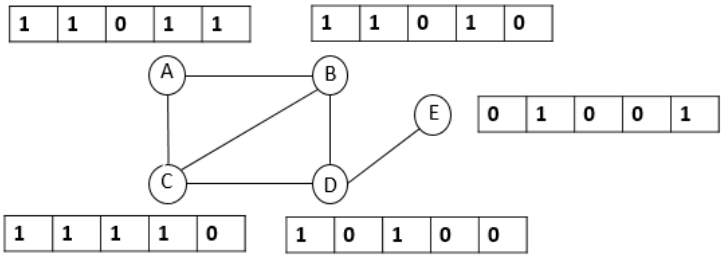


Figure 4.1. Attributed graph with 5 nodes

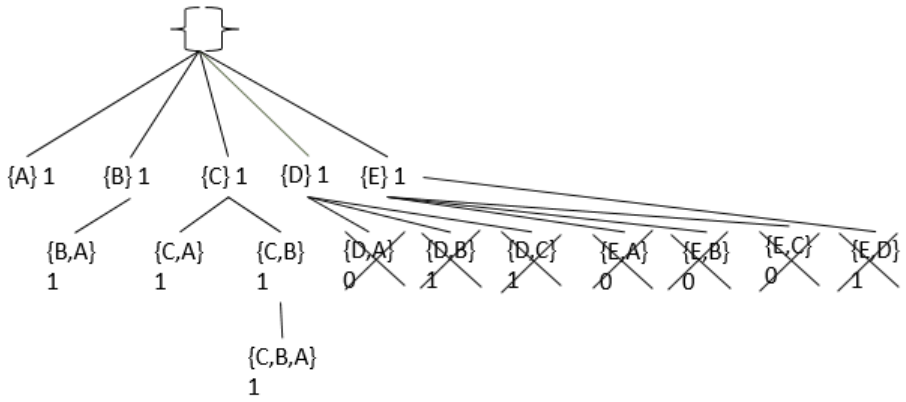


Figure 4.2. Dense module enumeration with density 0.3 and attribute length 3

Figure 4.2 shows how adding consistancy constraint can prune lot of child nodes and reduce the search space.

4.1. Data Processing

We Used actual protein interaction network as the graph input for DME. The gene (vertex) ID is mapped to actual EntrezID which is accepted universally to indicate specific gene. All the genes are marked as "1" if they are differentially expressed in the corresponding Cancer type , "0" otherwise. This binary files supply the profile information for genes in DME.

4.2. Finding Dense Cohesive Module

We run DME with the actual protein interaction network and attribute profile with different densities and attribute length. The density parameter varies from 0.6 to 0.9 and attribute length is changed from 5 to 10. We got modules of various lengths which fulfilled all the constraints. To prove the biological significance of the modules, we performed several tests.

4.2.1. Protein Interaction Data

The human protein-protein interaction data used in this experiment is obtained from the Biological General Repository for Interaction Datasets (BioGRID3.4.133). It stores gene and protein interactions data for human and all major model organism species. It is an open access database. It's current release is version 3.4.141 and stores information of 1,069,563 protein and genetic interactions of human species[1].

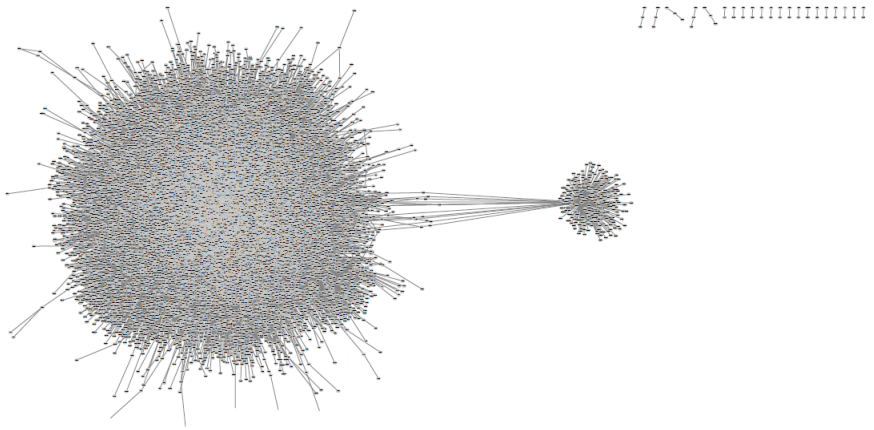


Figure 4.3. Protein-protein interaction network

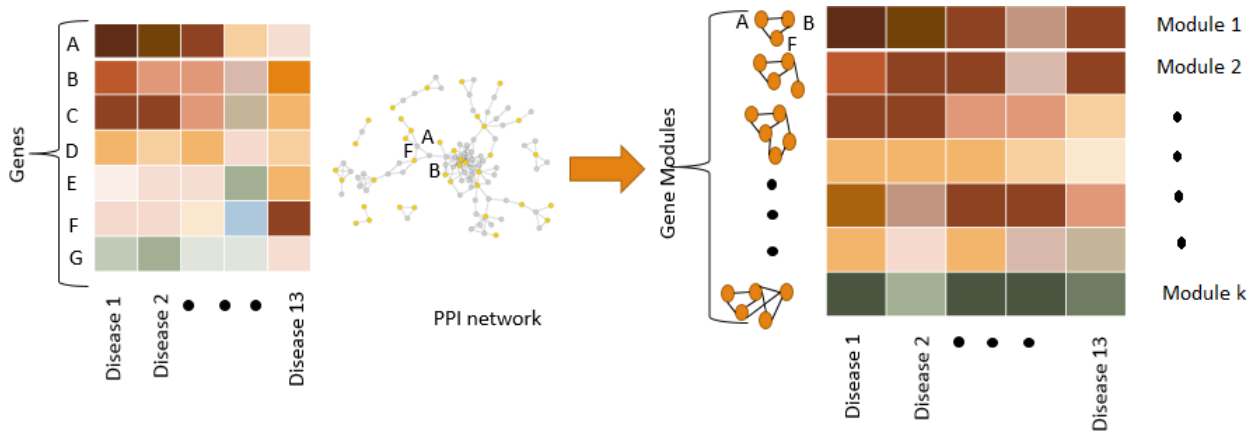


Figure 4.4. Gene modules generation from PPI network and gene dysregulation profiles

4.2.2. Partial Correlation

The partial correlation coefficient gives the correlation between two variables, say x and y, keeping a third variable, z, constant. This method tries to measure the similarity between x and y, over and above that caused by their common dependency on z. The partial correlation can be calculated as follows:

$$C_{xy.z} = \frac{C_{xy} - C_{xz.yz}}{\sqrt{(1-C_{xz}^2)}\sqrt{(1-C_{yz}^2)}}$$

The above formula can be expanded to condition on two variables as follows:

$$C_{xy.zw} = \frac{C_{xy.z} - C_{xw.z,yw.z}}{\sqrt{(1-C_{xw.z}^2)}\sqrt{(1-C_{yw.z}^2)}}$$

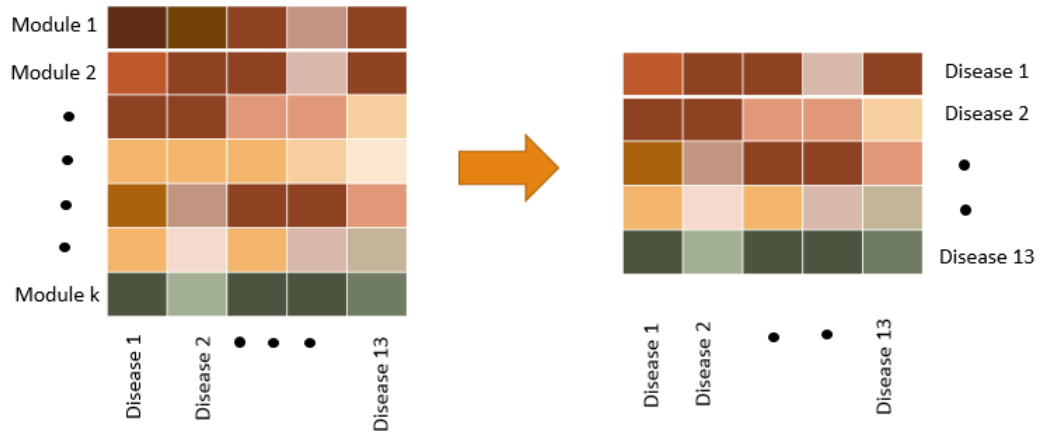


Figure 4.5. Calculating disease similarity from cohesive gene modules

Figure 4.4 and Figure 4.5 shows the overall process of generating cohesive dense module from protein interaction data and dyaregulated genes and finally using those modules to calculate disease similarity.

4.3. Disease Gene List

Great effort has been put on finding the genes associated to diseases. However, more and more evidences indicate that most human diseases cannot be attributed to a single gene but arise due to complex interactions among multiple genetic variants and environmental risk factors. Several databases have been developed storing associations between genes and diseases such as CTD (Comparative Toxicogenomics Database), OMIM and the NHGRI-EBI GWAS (Genome-wide Association Studies) catalog. Each of these databases focuses on different aspects of the phenotype-genotype relationship. In our experiment, we used DisGeNET database that currently contains 429,036 associations, between 17,381 genes and 15,093 human diseases [2].

5. RESULT

Protein complexes and pathways are responsible for most processes in the cell [12]. We can determine the irregularity of cell functions from the irregularity in the expression level of these complexes (Jin et al. [9]). In our experiment, we first collected data for 13 different types of cancer. Among all the genes, total 10396 genes are dysregulated in at least one of the thirteen Cancers. We ran the DME algorithm with PPI network data and DE gene profile. We altered the density and attribute length of modules from 0.6 to 0.9 and 5 to 10 respectively. For further experiments, we considered the modules we got as the result of DME algorithms with four or more cohesive genes.

Table 5.1. Number of modules and genes for attribute length 6 and density 0.6

# of dysregulated attribute	density	# of modules ≥ 4	Avg # of genes
5	0.6	29374	6.66
6	0.6	16999	6.67
7	0.6	3522	6.45
8	0.6	658	6.25
9	0.6	234	5.78
10	0.6	67	5.41

Table 5.1 shows the number of modules and average number of genes for varying parameters.

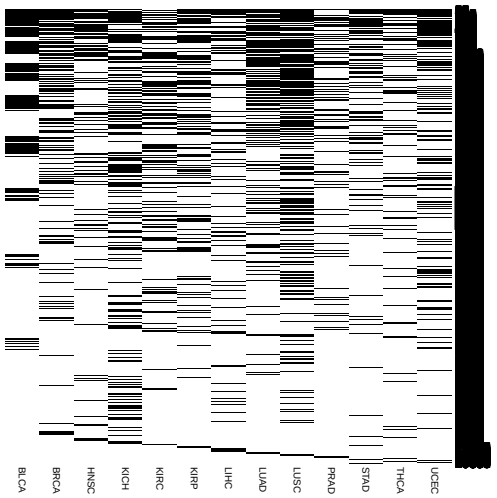


Figure 5.1. Heatmap of DE genes in Cancers

Figure 5.1 shows the heatmap of dysregulated genes in Cancer datasets.

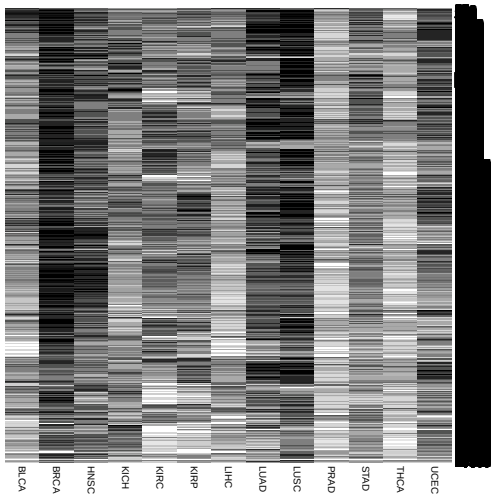


Figure 5.2. Heatmap of cohesive modules

Figure 5.2 shows the heatmap of dysregulated gene modules in Cancer Datasets.

We calculated Spearman partial correclation among diseases (Suthram et al. [15]). We considered the correlation significant if it has passed the Hypergeometric p-value threshold of 0.01. After filtering for P-value, we got 65 significant disease-diseae pair. From the help of database of disease genes (DisGeNET) we created a gene list those are known to be present in these 13 cancer types. A complete list of shared genes among diseases were created and Hypergeometric P-value was enlisted as well.

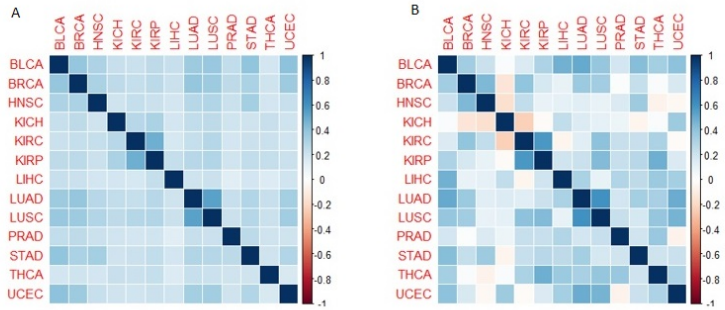


Figure 5.3. Disease-disease correlation without cohesive co-expression (A) Disease-disease correlation considering cohesive co-expression (B).

Figure 5.3 shows the disease pair similarity. Darker color indicates strong correlation between disease pair.

Table 5.2. Disease-disease correlation and shared genes between them

Disease 1	Disease 2	correlation	Disease 1 gene	Disease 2	shared genes	Hypergeometric P-value
KICH	KIRP	0.699	31	59	10	1.95E-20
BRCA	LUAD	0.773	369	847	133	1.94E-98
LUSC	UCEC	0.822	192	562	63	1.11E-54

Table 5.2 shows examples of disease pairs whose correlation are significant.

As we considered only one database to generate the disease gene list, so we did not get a good overlap of genes for diseases. That's why the result of Fisher test was not as good as we expected.

Table 5.3. One-sided Fisher's Exact Test on this table giving a p-value of 0.807

		Module based correlation		
		significant	Non-significant	Total
Shared disease genes	Significant	1	0	1
	Non-significant	62	15	77
	Total	63	15	78

Table 5.3 shows the contingency table for density 0.6, attribute length 6 and correlation 0.4 or greater.

To perform a different test, we downloaded data of all protein complexes available in mammal from most recent CORUM (Comprehensive Resource of Mammalian protein complexes) database. It has two different files containing the core complexes and containing all the protein complexes. The file contains information of complex id, complex names, synonyms , gene Entrez IDs etc. For the modules we got from DME algorithms, we calculated the overlap with complexes where the gene intersection length is three or more. From the CORUM dataset, we got total 2835

complexes, and we checked them with all the 16999 modules for attribute length 6 and density 0.6. We found 2033 significant overlap with the given complexes.

We additionally performed some biological enrichment analysis using the Database for Annotation, Visualization, and Integrated Discovery-DAVID [7]. In order to verify the significance of our results, we attempt to find enrichment of Gene Ontology process terms(GOTERMS) in our resulting patterns. We concluded, a pattern is enriched with a biological process function if that function is overrepresented in the genes from the pattern. In other words, the probability of there being a number of genes in a pattern that are involved in that process function by chance is statistically low, yet we have found them in that pattern. Because of this, we can say with a fair degree of certainty that those genes were not included in the pattern by chance the algorithm discovered. Rather there is a correlation between that biological function and the density and attribute similarity of the genes in the pattern. Whenever a cancer disease was present in the description of the enrichment analysis, we saved it and kept track how many times it occurred for a specific density and attribute length. Table 5.4 presents the top 10 cancer types that occurred in the modules for density 0.6 and attribute length 6.

Table 5.4. Top 5 occurring cancer types

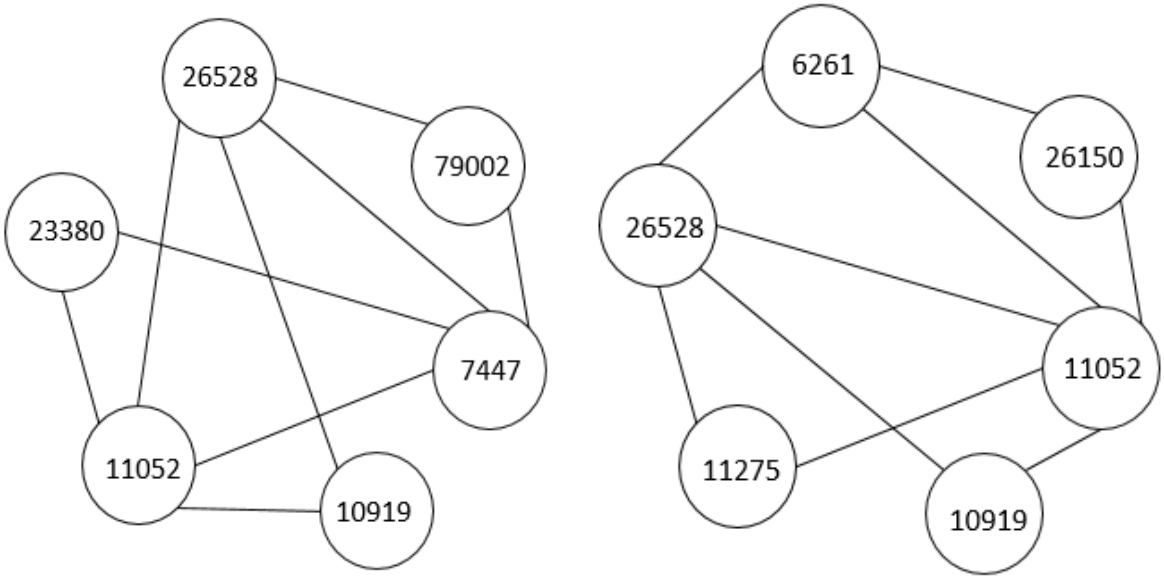
Cancer Name	**Count**
sensory system cancer	9110
ocular cancer	9110
female reproductive organ cancer	7414
hereditary breast ovarian cancer	7137
pancreatic cancer	6445

From the Table 5.4, we can clearly conclude that the modules that we found analyzing the dataset of 13 different types of cancer are also found in other cancer diseases as well. Among the top 5 cancers listed, only one was present in the actual dataset. In addition to cancer diseases, we also counted the occurrence of any disease to prove the significance of these modules in general disease state. Table 5.5 shows the top ten diseases for the same criteria. These two table support our statement that disease causing genes act together and are found in different types of diseases.

Table 5.5. Top 5 occurring diseases

Disease Name	Count
autosomal dominant disease	7702
thoracic disease	7678
breast disease	7678
bladder disease	3284
esophageal disease	3199

This analysis was performed for the modules having density 0.6 to 0.9 and attribute length 6 to 10. Among them most occurring cancer is **sensory system cancer** but for attribute length 8 or more **retinal cell cancer** is also frequently occurring one.



(a) Pancreatic Cancer module (b) Embryoma Disease module

Figure 5.4. Modules of most occurring cancer and disease

Figure 5.4 shows two examples of Cohesive Dense Modules.

6. CONCLUSION AND FUTURE WORK

In this paper, we tried to find similarity between diseases at the genomic level. Instead of single gene, we focused on the group of cohesive genes sharing similar properties. We proposed a method to find disease similarity considering gene interaction network and attribute profiles. We have showed that disease pair having strong correlation shares a number of cohesive genes. Our method used DME algorithm which ensure all the dense patterns extracted are maximal and are not redundant. Finally, we performed enrichment analysis to assess the accuracy of our results. From researchers point of view, finding similarity between diseases is really important because if we find out that same modules of genes are responsible for multiple diseases, we can treat more and more diseases with same medicines. Even it is possible that we can cure diseases with existing medicines.

In future, this research can be extended and results can be improved. If we incorporate more and more information like combining gene expression data, environmental factors and the actual values of the gene attributes, we might get better result. With actual values of gene expression profile we can alter the parameters and observe its impact on generating the modules. Considering attribute length less than five can also enrich the result significantly. This experiment can be done and might outperform for different kinds of disease dataset other than cancer.

REFERENCES

[1] BioGRID— database of protein, chemical, and genetic interactions. Accessed: 2016-09-08.

[2] DisGeNET-a database of gene-disease associations. Accessed: 2016-09-08.

[3] Michael J. Berry and Gordon Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer Support*. John Wiley & Sons, Inc., New York, NY, USA, 1997. ISBN 0471179809.

[4] A M Cohen and W R Hersh. A survey of current work in biomedical text mining. *Brief Bioinform*, 6(1):57–71, 2005.

[5] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. *From Data Mining to Knowledge Discovery in Databases*. Association for the Advancement of Artificial Intelligence (www.aaai.org), 1996.

[6] Elisabeth Georgii, Sabine Dietmann, Takeaki Uno, Philipp Pagel, and Koji Tsuda. Enumeration of condition-dependent dense modules in protein interaction networks. *Bioinformatics*, 25(7):933–940, 2009.

[7] Da Wei Huang, Brad T Sherman, and Richard A Lempicki. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat. Protocols*, 4(1):44–57, dec 2008.

[8] Wei Jiang, Ramkrishna Mitra, Chen-Ching Lin, Quan Wang, Feixiong Cheng, and Zhongming Zhao. Systematic dissection of dysregulated transcription factor-miRNA feed-forward loops across tumor types. *Briefings in bioinformatics*, (October):bbv107–, 2015.

[9] Ruoming Jin, Scott McCallen, Chun-Chi Liu, Yang Xiang, Eivind Almaas, and Xianghong Jasmine Zhou. Identifying dynamic network modules with temporal and spatial constraints. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 214:203–214, 2009.

[10] L. Kaufman and P. Rousseeuw. Clustering by means of medoids. *Statistical Data Analysis Based on the L1-Norm and Related Methods*, pages 405–416, 1987.

[11] Victor E Lee. *Managing and Mining Graph Data (Advances in Database Systems)*. 2010. ISBN 9781441960443.

[12] Trudy Fc Mackay and Jason H Moore. Why epistasis is important for tackling complex human disease genetics. *Genome medicine*, 6(6):124, 2014.

[13] F. Moser, R. Colak, a. Rafiey, and Martin Ester. Mining cohesive patterns from graphs with feature vectors. *Proc Int SIAM Conf on Data Mining*, pages 593–604, 2009.

[14] Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. *ACM SIGMOD Record*, 25(2):1–12, 1996.

[15] Silpa Suthram, Joel T. Dudley, Annie P. Chiang, Rong Chen, Trevor J. Hastie, and Atul J. Butte. Network-based elucidation of human disease similarities reveals common functional modules enriched for pluripotent drug targets. *PLoS Computational Biology*, 6(2):1–10, 2010.

[16] Ah-Hwee Tan. Text Mining: The state of the art and the challenges. *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, 8:65–70, 1999.

[17] Andrzej Trybulec. Enumerated Sets. *Formalized Mathematics*, (1), 1990.

[18] Takeaki Uno, Tatsuya Asai, Yuzo Uchida, and Hiroki Arimura. LCM: An Efficient Algorithm for Enumerating Frequent Closed Item Sets. *Fimi*, 90, 2003.

[19] Mohammed J. Zaki and Karam Gouda. Fast vertical mining using diffsets. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, page 326, 2003.

[20] Mohammed J Zaki and Wagner Meira Jr. *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

[21] Mohammed Javeed Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara, and Wei Li. New Algorithms for Fast Discovery of Association Rules. *3rd Intl Conf on Knowledge Discovery and Data Mining*, 20(651):283–286, 1997.