INVESTIGATING GENE RELATIONSHIPS IN MICROARRAY EXPRESSIONS: APPROACHES USING CLUSTERING ALGORITHMS

A Thesis

Presented to

The Graduate Faculty of The University of Akron

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Mohammad Shabbir Hasan

August, 2013

INVESTIGATING GENE RELATIONSHIPS IN MICROARRAY EXPRESSIONS:

APPROACHES USING CLUSTERING ALGORITHMS

Mohammad Shabbir Hasan

Thesis

Approved:

_____
Advisor
Dr. Zhong-Hui Duan

_____
Committee Member
Dr. Yingcai Xiao

_____
Committee Member
Dr. Kathy J. Liszka

Accepted:

_____
Department Chair
Dr. Yingcai Xiao

_____
Dean of the College
Dr. Chand K. Midha

_____
Dean of the Graduate School
Dr. George R. Newkome

_____
Date

ABSTRACT

DNA Microarray technology provides a convenient way to investigate expression levels of thousands of genes in a collection of related samples during different biological processes. Researchers from diverse disciplines such as computer science and biology have found it interesting as well as meaningful to group genes based on the similarity of their expression patterns. Different clustering algorithms such as hierarchical clustering, k-means clustering, self-organizing maps have been applied to group of genes with similar expression patterns. However these traditional clustering algorithms suffer from various limitations. Beside these clustering algorithms, there are other algorithms to group similar items together. Ford Fulkerson algorithm which is based on maximum flow – minimum cut approach is one of them and it is widely used for community discovery in web graphs. The aim of this research work is, to group genes with similar expression pattern using two different approaches: one is the k-means clustering combined with hierarchical clustering and the other is maximum flow – minimum cut approach in association with Dijkstra's algorithm to select source and sink nodes.

We use a publicly available microarray data on Adenocarcinoma which is the most common type of non-small-cell cancers. This dataset is available in the Gene Expression Omnibus which is a public domain functional genomics data repository. This

dataset contains samples of five different groups: normal tissue, tissues with EGFR mutation, tissues with KRAS mutation, tissues with EML4-ALK fusion and tissues with EGFR, KRAS, EML4-ALK negative cases. We investigate a number of representative genes from the group of normal tissue and from the group of KRAS mutation tissues which is also termed as KRAS positive groups in this study. We clustered the genes for both of these groups. Finally we used Gene Ontology database to find changes in the enrichment of molecular functions of the genes contained in each cluster discovered by the above mentioned approaches for both normal and KRAS positive dataset.

We discovered that both of these approaches can group genes with similar expression pattern together and hence we proposed that these approaches can be used in future for clustering microarray data.

ACKNOWLEDGEMENTS

I would like to give my sincere gratitude to my advisor Dr. Zhong-Hui Duan for her constructive ideas, feedbacks and support since the very beginning of this research work. Her nice and friendly supervision have made it possible to achieve the expected outcome from this thesis. I feel very lucky to get the opportunity to work with her and to enhance the knowledge required for doing bioinformatics research.

I would like to give thanks to my thesis committee members Dr. Yingcai Xiao and Dr. Kathy Liszka for their taking valuable time and providing insightful feedbacks to improve this thesis. I really appreciate their contributions.

Most importantly, my heartiest thanks to my parents and sisters for their blessings and moral support along the way. I would like to extend a special thanks to my friends of the University of Akron for their support.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

1.1 Overview

DNA microarray technology which has become a very useful tool to get information for diagnosis of different diseases often requires algorithms to analyze DNA microarray datasets accurately. Clustering algorithms play an important role in gene analysis by separating a dataset of heterogeneous genes into homogeneous groups containing similar genes. It helps to analyze a group of genes instead of analyzing each one individually. After getting appropriate clusters, researchers can further investigated the clusters to find distinct pattern for each cluster as well as find more information about functional similarities and gene interactions. A large number of algorithms have been developed for clustering DNA microarray data so far. Tavazoie et al. applied the k–means clustering algorithm for yeast data [1] and Luo et al. used hierarchical clustering algorithm in genomic research [2]. Unfortunately both of these algorithms suffer from some limitations such as the performance of k-means clustering depends of how efficiently the initial number of cluster is determined and hierarchical clustering algorithm requires high computational complexity to discover optimal cluster.

There is also another clustering algorithm which uses maximum flow minimum cut approach which is mostly used in clustering web graph for web community discovery [3]. Algorithms based on this approach are relatively fast and simple, and have been used in the past for clustering web graphs [4 –5]. We believe this approach can be used for gene clustering as well.

In this research work, we propose two approaches for gene clustering which includes an algorithm that combines both hierarchical clustering and k-means clustering. The other approach uses the maximum flow minimum cut algorithm. The first approach first uses the hierarchical clustering to decide the initial number of clusters and then feed this information to k-means clustering to obtain the final clusters. The second approach requires a weighted graph. So we represented the DNA microarray data by a weighted graph where the genes are represented as the nodes of the graph and the weight of the edges are represented by the Pearson Correlation coefficient value between the corresponding genes. The graph can be partitioned into two disjoint sub graphs by each graph cut. The algorithm is applied recursively on the sub graphs until no new cluster can be discovered. After getting the new clusters from these two approaches, at the end, we explored the change in enrichment of molecular functionalities of the genes of each cluster for normal tissue and cancer tissue by using Gene Ontology (GO) annotations.

1.2 Research Goals

The goals of this research work are listed below:

1. Implementing a combined clustering algorithm that uses hierarchical clustering and k-means clustering algorithm.
2. Implementing a clustering algorithm that uses maximum flow minimum cut algorithm.
3. Developing software to be used as a platform for the proposed approaches.
4. Finding an appropriate dataset to be used as the input data source.
5. Analyzing the change in enrichment of the genes' molecular functionalities genes in the clusters discovered by the proposed approaches using Gene Ontology annotations.

1.3 Organization of the thesis

- Chapter 1 introduces the necessity of gene clustering as well as the goals of this research work.
- Chapter 2 explains the background necessary to understand the materials and methods used here. This chapter discusses about the algorithms used along with a brief discussion about the Pearson correlation coefficient which we use as the similarity matric, gene expression and gene ontology.
- Chapter 3 gives an overview of previous research works.

- Chapter 4 explains the materials and methods used for clustering genes. This chapter describes in detail about the data set used and also how the algorithms were used for clustering the genes given in the data set.
- Chapter 5 discusses the results of this research work and analyzes the result using Gene Ontology annotations.
- Chapter 6 concludes the research work presented here.

CHAPTER II

BACKGROUND

This chapter describes necessary background concepts related to this work. A brief description of the gene expression is given. It is followed by the Pearson Correlation Coefficient which is used as the similarity score among the genes is given in this chapter. K-means and hierarchical clustering algorithms are also explained in detail along with Dijkstra's shortest path algorithm and Ford Fulkerson algorithm. We also discuss about Gene Ontology which is used for gene enrichment analysis.

2.1 Gene Expression

In gene expression, gene products such as proteins or RNA are created from the inheritable information contained in a gene [6]. So far traditional molecular biology has focused on studying individual genes in isolation for determining gene functions, but it is not suitable for determining complex gene interactions as well as explaining the nature of complex biological processes. For this purpose, examining the expression pattern of a large number of genes in parallel is required [7]. DNA microarray technology which is one of the most important tools now-a-days for the analysis of gene expression has made it possible to view thousands of genes expression levels in parallel [8]. It is believed that a group of genes with similar gene expressions are likely to have related gene functions

[9]. Hence identifying genes with similar expression levels in different phases of the cell cycle or in different environmental conditions is an important task.

2.2 Pearson Correlational Coefficient

The Pearson correlation coefficient developed by Karl Pearson from a related idea introduced by Francis Galton [10 - 11] is a measure of the correlation between two variables X and Y, giving a value between +1 and −1 inclusive. It is widely used as a measure of the strength of linear dependence between two variables. In this study, each variable represents the expression level of a gene which is also referred as an object in this thesis.

Let's consider the expression levels of gene X and Y, $X = \{x_1, x_2, x_3, \ldots\ldots\ldots, x_n\}$ and $Y = \{y_1, y_2, y_3, \ldots\ldots\ldots, y_n\}$ where x_i is the expression level of gene x in sample i. The Pearson correlation coefficient between these genes can be defined as

$$r = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma_x}\right)\left(\frac{y_i - \bar{y}}{\sigma_y}\right) \tag{2.1}$$

where $\bar{x}$ is the average of values in X, and σ_x is the standard deviation of these values.

The values for Pearson correlation coefficient range from -1 to 1. If a linear equation describes the relationship between x and y perfectly and all data points lies on a line, with the correlation value 1 it means y increases as x increases and the correlation value -1 means the completely opposite thing i.e., y decreases as x increases. With a

correlation value 0 it means x and y are completely uncorrelated and there is no linear relation between them.

There are many ways of theorizing the correlation coefficient. If we consider a scatterplot of the values of x against y i.e. pairing x_1 with y_1, x_2 with y_2 and so on, then the Pearson correlation coefficient r reports how well we can fit a line to the values as shown in Figure 2.1.

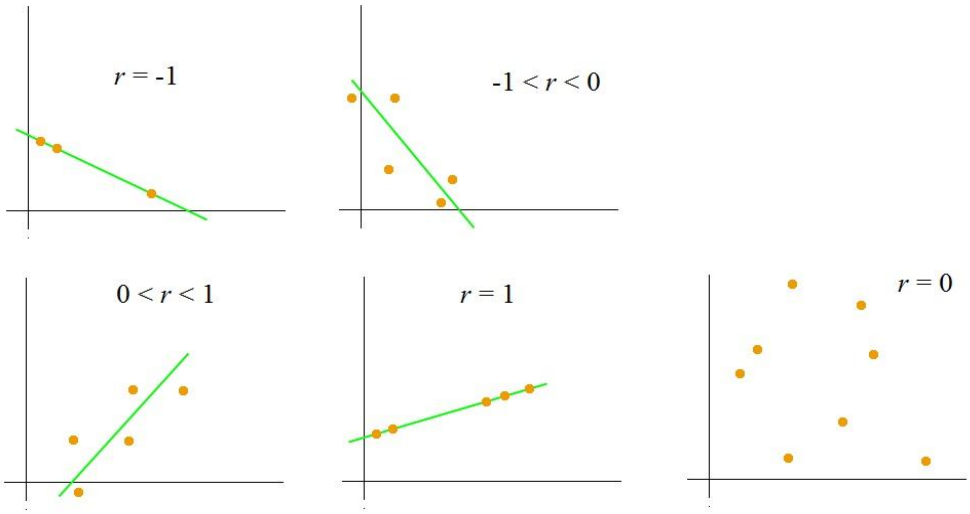

Figure 2.1: Scatter diagrams with different values of Pearson correlation coefficient (r)

2.3 Calculating the distance matrix from Pearson Correlation Coefficient

The first step in the hierarchical clustering discussed later is to calculate the distance between all pairs of object to be clustered. This distance is the opposite of the similarity. The distance between X and Y can be calculated using the following equation:

$$distance = 1.0 - r \tag{2.2}$$

where r is the value of Pearson correlation coefficient.

2.4 K-means Clustering Algorithm

K-means clustering is a well-known method for cluster analysis which partitions expression levels of n genes into k clusters where each gene belongs to the cluster with the nearest mean. Normally, the existing heuristic algorithms are used until the clustering converges quickly to a local optimum.

Let's consider a set of expression value of n genes ($x_1, x_2, \ldots, x_n$). It is a vector with d-dimension where d is the number of samples in the dataset and k-means clustering algorithm partitions the n genes into k sets ($k \leq n$) $S = \{S_1, S_2, \ldots, S_k\}$ until the criterion function converges. Typically, the square-error criterion given in equation (2.3) is used to measure whether an optimal is reached

$$\mathrm{E} = \sum_{i=1}^{k} \sum_{x_j \epsilon S_i} \left\| x_j - \mu_i \right\|^2 \tag{2.3}$$

where E is the sum of the square error for all genes in the data set and μ_i is the mean of points in S_i. Here for each gene in each cluster, the distance from the genes to its cluster center is squared and, the squares are summed up. This criterion tries to make the resulting k clusters as compact and as separate as possible. The clustering process is summarized in Figure 2.2.

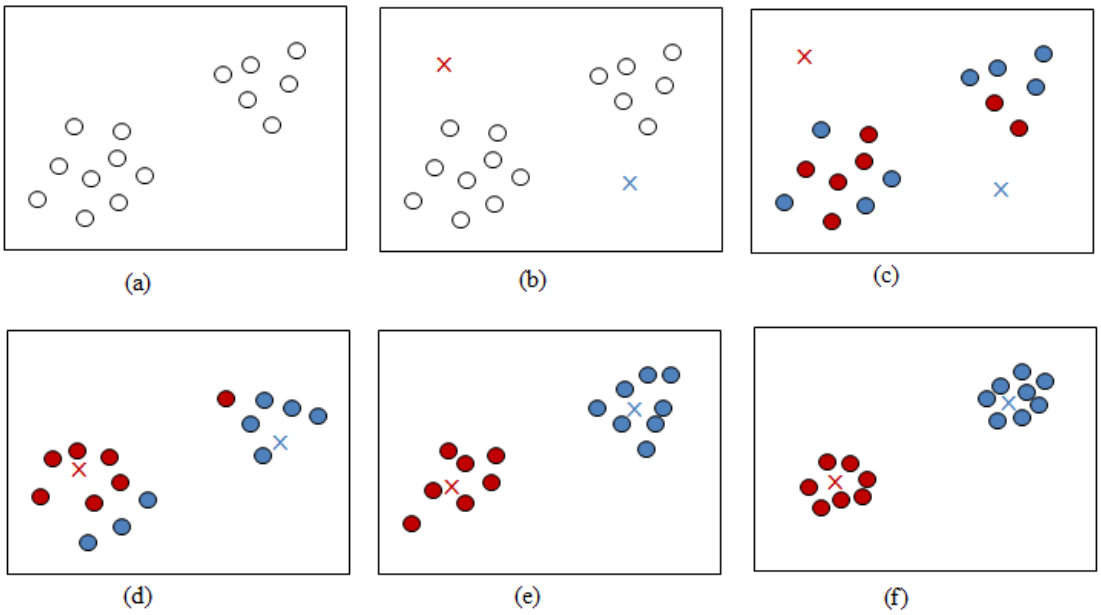

Figure 2.2: Illustration of the k-means clustering algorithm.

In Figure 2.2 a circle represents a gene and cross represents the centroid of a cluster of genes. Here step (a) shows the original dataset containing the genes. Step (b) shows the initial cluster centroids selected randomly. Steps (c) to (f) show the illustration of running two iterations of k-means. In every iteration, each gene is assigned to the closest cluster centroid, shown by painting the gene the same color as the cluster centroid to which is assigned. Then each cluster centroid is moved to the mean of the points assigned to it.

The Algorithm for k-means clustering is given below:

Input:

- *k*: the number of clusters
- *S*: a dataset containing *n* genes

Output: A set of *k* clusters

Steps:

1. arbitrarily choose *k* genes from *S* as the initial cluster centers;
2. repeat
    a. (re) assign each gene to the cluster to which the gene is the most related, based on the means value of the genes in the cluster;
    b. update the cluster means, i.e., calculate the mean value of the genes for each cluster;
3. until no change in the clusters take place.

Advantages of using this technique are:

1. It is computationally faster than other clustering algorithm (ex. hierarchical clustering) with a large number of variables.
2. It produces tighter cluster than hierarchical clustering.

Disadvantage of using this technique is:

It is difficult to select what should be the value of *k*.

2.5 Hierarchical Clustering

In gene clustering, hierarchical clustering is a method for cluster analysis which builds a hierarchy of clusters. This clustering method organizes genes in a tree structures based on their relation. The basic idea is to assemble a set of genes into a tree, where

genes are joined by very short branches if they have very high similarity to each other and by increasingly long branches as their similarity decreases.

The approaches for hierarchical clustering can be classified into two groups: agglomerative and divisive. The agglomerative approach is a ―bottom up" approach where each gene starts in its own cluster and pairs of clusters are merged as one moves up the hierarchy. On the other hand, divisive approach is a ―top down" approach where all genes starts in one cluster and splits are performed recursively as one moves down the hierarchy. Figure 2.3 summarizes both approaches.

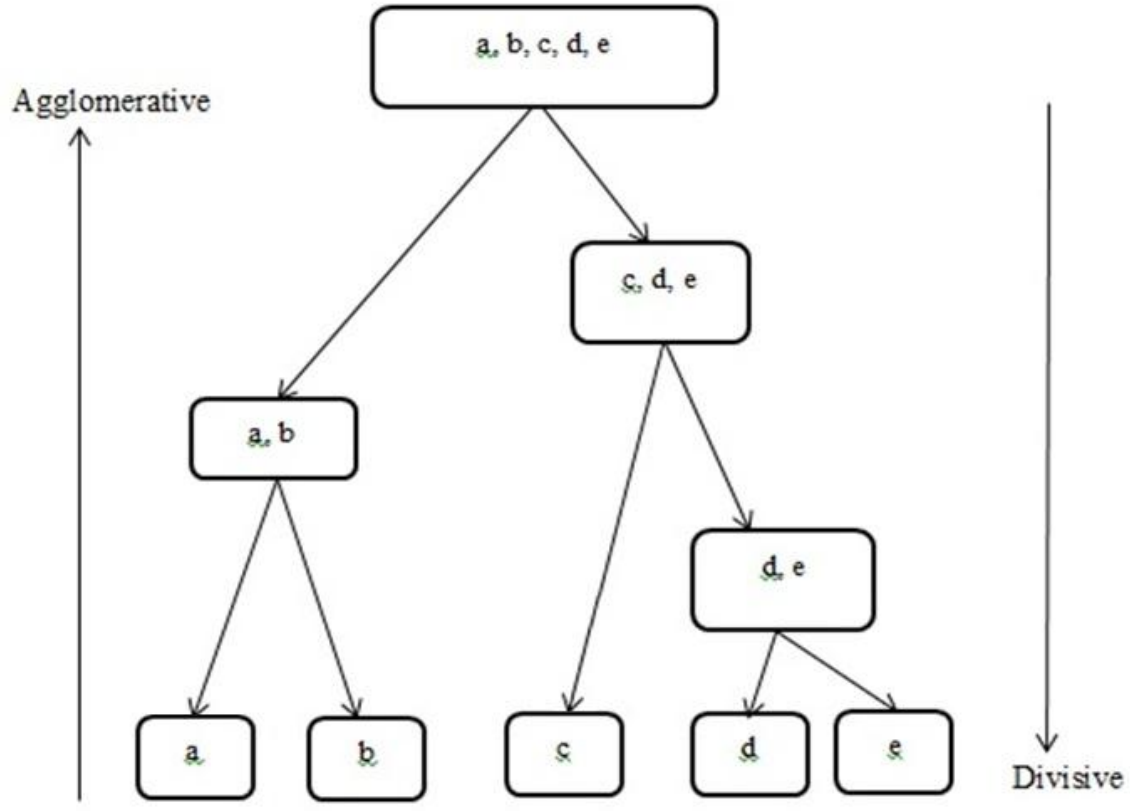

Figure 2.3: Agglomerative and Divisive approaches for hierarchical clustering.

In hierarchical clustering, the first step is to calculate the distance matrix between the genes in the data set. The clustering starts once this matrix of distances is computed. The agglomerative hierarchical clustering technique consists of repeated cycles where the two closest genes having the smallest distance are joined by a node. In this study this new node has been termed as pseudo node. The two joined genes are removed from the list of genes being processed and replaced by the pseudo node that represents the new branch. The distances between this pseudo node and all other remaining genes are computed, and the process is repeated until only one node remains.

There are a variety of ways to compute distances while dealing with pseudo node: centroid linkage, single linkage, complete linkage, and average linkage.

2.5.1 Centroid Linkage

In centroid linkage clustering, an average expression profile also knows as centroid is calculated in two steps. First, the mean in each sample of the expression profiles is calculated for all genes in a cluster. Then, distance between the clusters is measured as the distance between the average expression profiles of the two clusters.

2.5.2 Single Linkage Clustering

In single linkage clustering, distance between two clusters of genes is calculated as the minimum distance between all possible pairs of genes, one from each cluster. This method has an advantage that it is insensitive to outliers. This method is also known as the nearest neighbor linkage. Unlike centroid linkage clustering, once the distance matrix

is known, no further distance need to be calculated in single linkage clustering. Hence, single linkage clustering is much faster and more memory efficient.

2.5.3 Complete Linkage Clustering

In complete linkage clustering, distance between two clusters of genes is calculated as the maximum distance between all possible pairs of genes, one from each cluster. The disadvantage of this method is that it is sensitive to outliers. This method is also known as the farthest neighbor linkage. In complete linkage clustering, once the distance matrix is known, no more distance need to be calculated.

2.5.4 Average Linkage Clustering

In average linkage clustering, distance between two clusters of genes is calculated as the average of distances between all possible pairs of genes in the two clusters. In bioinformatics this one is also known as UPGMA (Unweighted Pair Group Method with Arithmetic Mean) which is used to produce guide trees for more sophisticated phylogenetic reconstruction algorithms.

Figure 2.4 shows the algorithms discussed above to find distance between two clusters of genes.

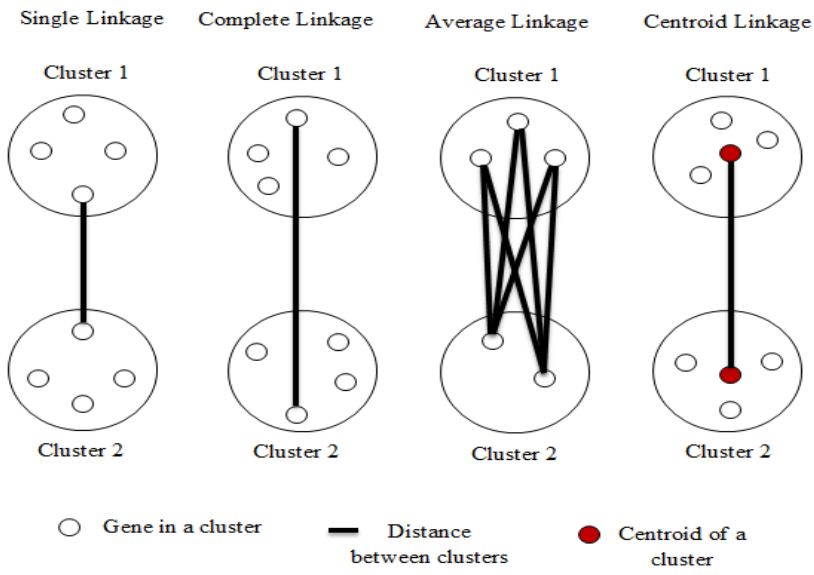

Figure 2.4: Different algorithms to find distance between two clusters

Figure 2.5 illustrates an example of hierarchical clustering that uses single linkage algorithm for calculating distance between two clusters. [12]

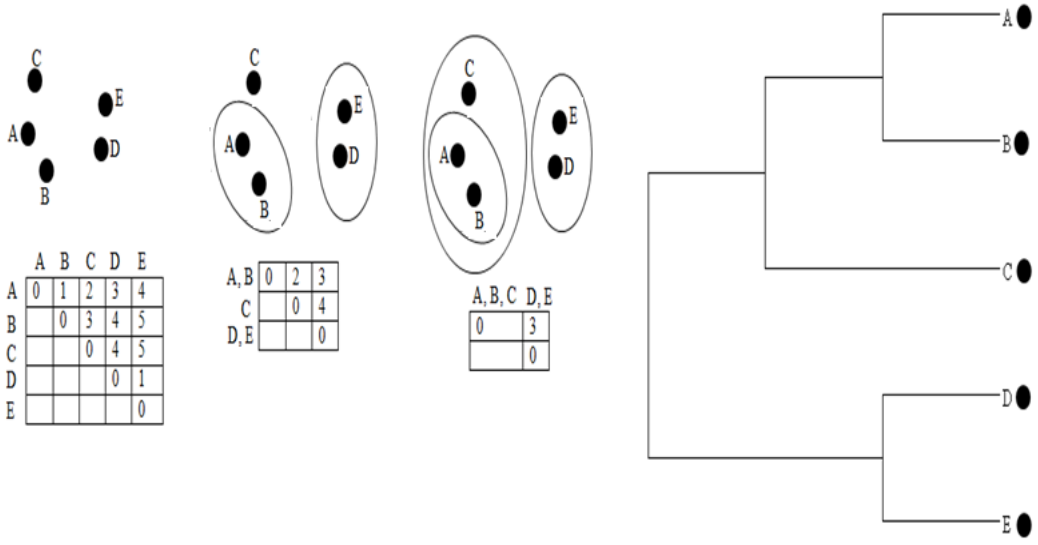

Figure 2.5: Illustration of hierarchical clustering with single linkage algorithm

In Figure 2.5, black circles represent the genes in the dataset. There are five genes and the distances between the genes are given in the first table. At step 1, genes that are close to each other and the distances are re-calculated by using the single linkage algorithm. This steps repeats until all genes are grouped into one cluster. This procedure is shown in the dendrogram and length of a branch represents the distance between genes and clusters.

Advantages of using hierarchical clustering are:

1. Does not require the number of clusters to be known in advance
2. Computes a complete hierarchy of clusters.

Disadvantage is:

1. There is no automatic discovering of ~~o~~ptimal" clusters.

2.6 Dijkstra's Algorithm

Let's consider a weighted directed graph $G = (V, E)$ where V is the set of genes and E is the set of edges in G. Here also consider that all edge weights are nonnegative i.e. , $w(u,v) \geq 0$ for each edge $(u, v) \in E$. In this study, the weights represent the distance between the genes. Dijkstra's algorithm is a graph search algorithm that solves the single source shortest path problem for G. This algorithm is often used in GPS technology to find shortest route.

In the implementation shown below, Dijkstra's algorithm maintains a set S of genes whose final shortest-path weights from the source s have already been determined. The algorithm repeatedly selects the vertex (gene) $u \in V - S$ with the minimum shortest-path estimate, adds u to S, and relaxes all edges leaving u. This implementation uses a min-priority queue Q of vertices, keyed by their distance values, d [13]

DIJKSTRA (G, w, s)

1. INITIALIZE-SINGLE-SOURCE (G, s)
2. $S = \Phi$ // Initializes the set S to the empty set
3. $Q \longleftarrow V$ // Initializes the min-priority Q queue to contain all the vertices (genes) in V
4. **while** $Q \neq \phi$ // Until Q is not empty
5. u = EXTRACT-MIN (Q) // extract a vertex (gene) u from $Q = V$-S
6. $S = S \cup \{u\}$ // add the extracted vertex (gene) u to set S
7. **for** each vertex v which is adjacent to u
8. RELAX (u,v,w)

INITIALIZE-SINGLE-SOURCE (G, s)

1. **for** each vertex v $\in$ V
2. $v.d = \infty$ // initially the shortest-path estimate for node v is infinite
3. $v.\pi$ = NIL // $v.\pi$ means predecessor attribute of v
4. $s.d = 0$ // initially the shortest-path estimate for the source node s is zero

RELAX (u,v,w)

1. **if** $v.d > u.d + w(u,v)$ // if any path found which is shorter than the previously found shortest path
2. $v.d = u.d + w(u,v)$ // update the shortest-path estimate of v
3. $v.\pi = u$ // update the predecessor attribute of v

The process of relaxing an edge (u,v) in the function RELAX consists of testing whether the shortest path to v found so far can be improved by going through u and if any such path found, then update $v.d$ and $v.\pi$. This process may decrease the value of the shortest-path of v i.e. $v.d$ and update the predecessor attribute of v i.e. $v.\pi$.

Figure 2.6 illustrates an example of the Dijkstra's algorithm. In this figure s is the source node, dashed edges indicate predecessor values. In this figure black vertices are in the set S, and white vertices are in the min-priority queue $Q = V - S$.

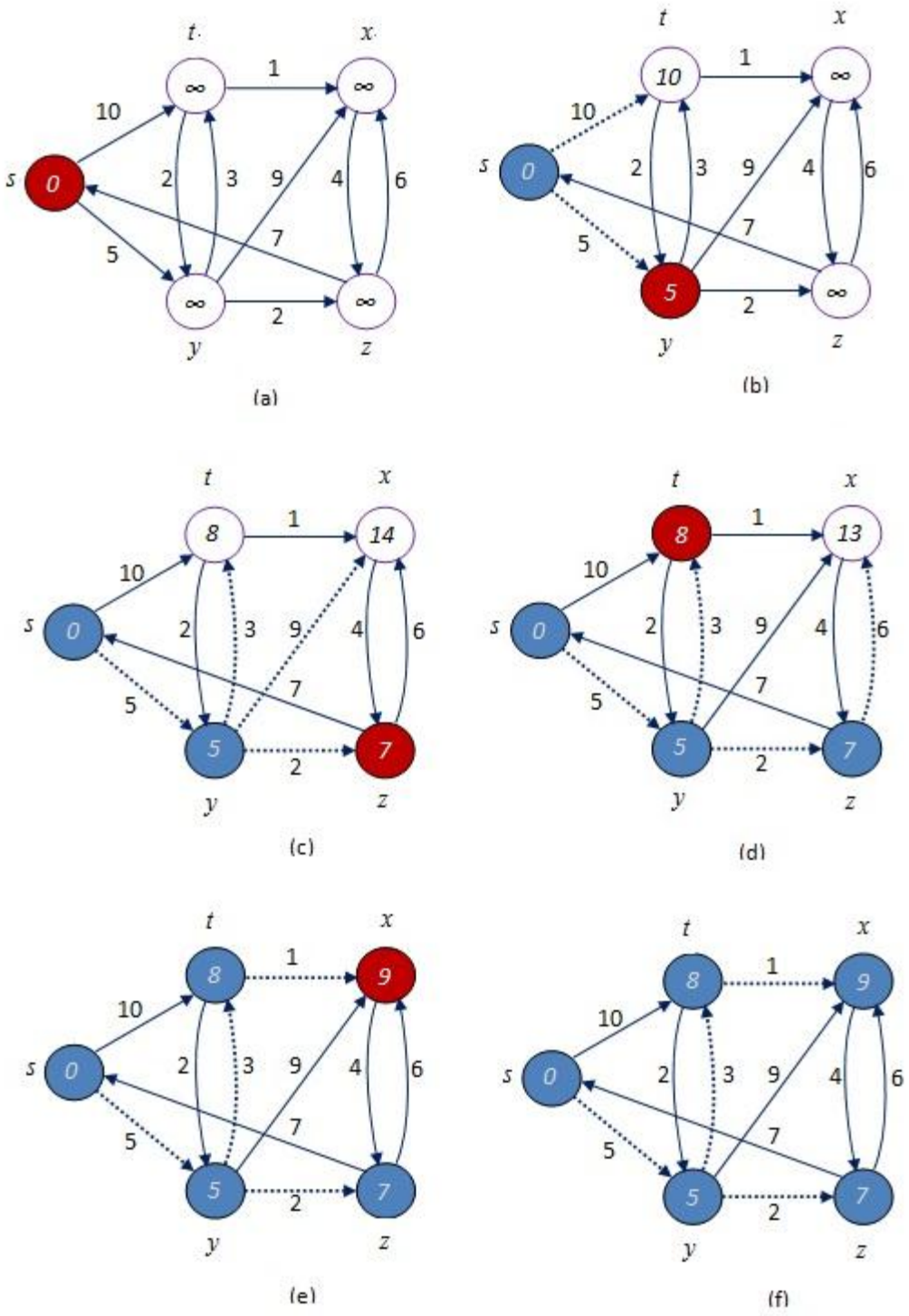

Figure 2.6: Example of Dijkstra's algorithm

2.7 Maximum-flow Minimum-cut Theorem

Maximum-flow Minimum-cut theorem states that in a flow network, the maximum amount of flow passing from the source node to the sink node is equal to the minimum capacity. If this minimum capacity is removed from the network in a specific way, it causes the situation that no flow can pass from the source to the sink node.

Let's consider $N = (V, E)$ be a directed graph which represents a network and s and t are the source and the sink node of N respectively. The capacity of an edge is $c(u,v)$ that represents the maximum amount of flow that can pass through that edge (u,v). A flow is $f(u,v)$ which is subject to the following constraints:

1. $0 \leq f(u,v) \leq c(u,v)$ for all $u,v \in V$
2. $\sum_{v \in V} f(v,u) = \sum_{v \in V} f(u,v)$ for all $u \in V - \{s, t\}$

The maximum-flow problem is to maximize $|f|$ where $|f|$ is defined by $|f| = \sum_{v \in V} f(s,v)$ where s is the source of N. The purpose is to route as much flow as possible from source node s to sink node t.

The minimum-cut problem is to minimize $c(S, T)$ where $c(S, T)$ is defined by $c(S, T) = \sum_{(u,v) \in S \times T} c(u,v)$ and the purpose is to determine S and T such that the capacity of S-T cut is minimal. Here S and T are two disjoint sets and $S \cup T = V$.

2.7.1 Ford Fulkerson Algorithm

The Ford-Fulkerson algorithm computes the maximum flow in a flow network. The idea behind the algorithm is as long as there is a path from the source to sink node, with

available capacity on all edges in the path, flow should be sent along one of these paths. Then another path is found and so on. Here a path with available capacity is known as augmenting path.

Let's consider $N = (V, E)$ be a directed graph which represents a network and s and t are the source and the sink node of N respectively. Here V is set of vertices and E is the set of edges in the graph. The residual network of N is a network $G_f(V, E_f)$ with capacity $c_f(u,v) = c(u,v) - f(u,v)$ and no flow.

FORD-FULKERSON(G, c, s, t)

// G is the graph, c contains the capacity for all edges, s is the source node and t is the sink node

1. $f(u, v) = 0$ for all edges (u, v)
2. **while** there is a path p from s to t is G_f such that $c_f(u,v) > 0$ for all edges $(u,v) \in p$ // G_f is the residual network and c_f is the residual capacity.
3. find $c_f(p) = \min \{ c_f(u,v): (u,v) \in p \}$
4. **for** each edge $(u,v) \in p$
5. $f(u, v) = f(u, v) + c_f(p)$ // send flow along the path
6. $f(v, u) = f(v, u) - c_f(p)$ // the flow might be returned later

The path in step can be found using Breadth First Search (BFS) or Depth First Search (DFS). Figure 2.7 shows an example of the Ford-Fulkerson algorithm.

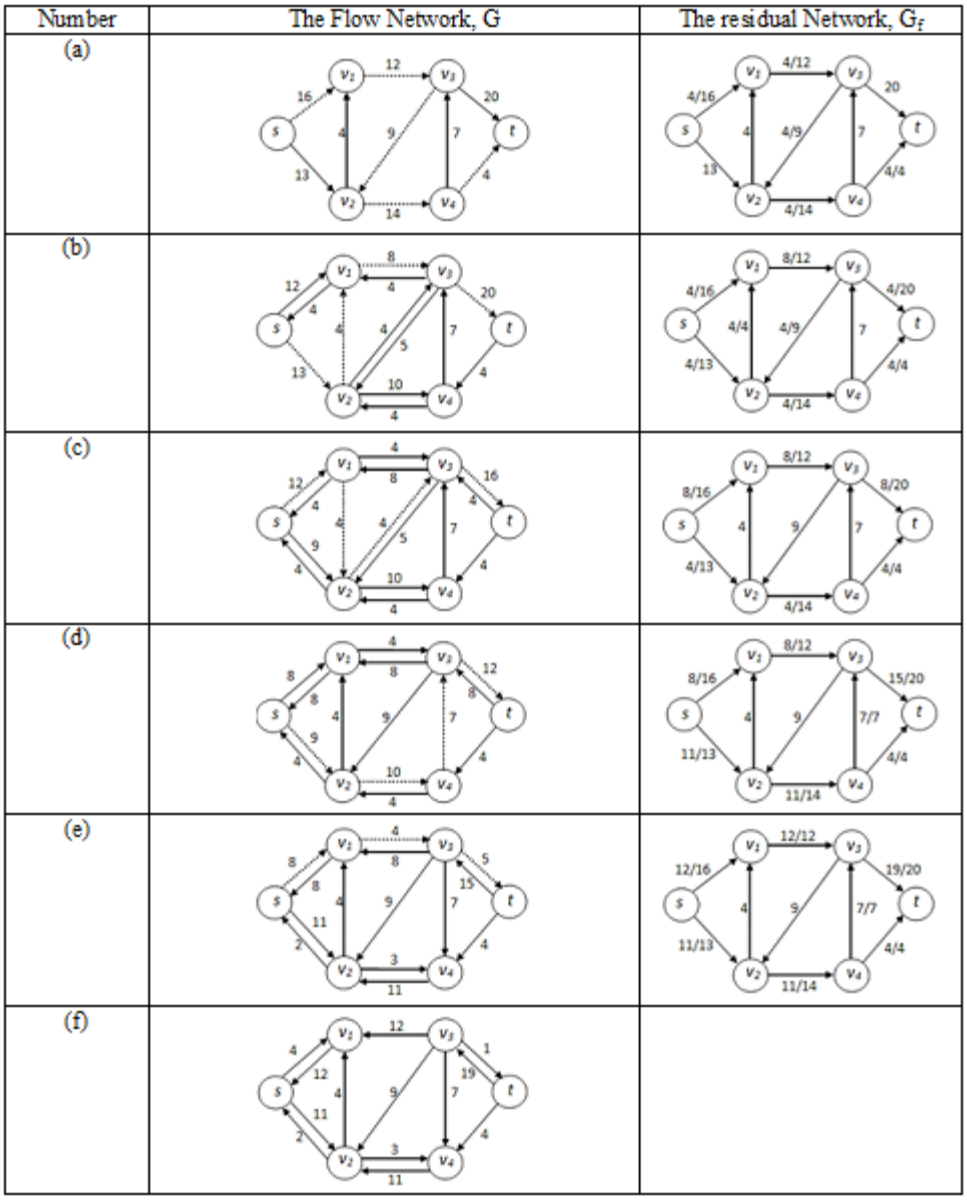

Figure 2.7: An example of Ford-Fulkerson algorithm

In Figure 2.7 steps (a) to (e) show the successive iteration of the while loop. The residual network G_f is shown at the left side of each part with a dashed augmenting path p. The right side of each part shows the new flow f resulted from the augmenting path f *by* f_p. Here the network shown in (a) is the input network and

the network shown in (f) is the residual network with no augmenting path. Therefore the flow *f* shown in (e) is a maximum flow and the value of the maximum flow is 23.

2.8 Gene Ontology

Gene Ontology (GO) is a set of associations relating biological phrases to specific genes. GO is designed to encapsulate the known relationships between biological terms and genes that are instances of these terms. It is helpful for biologists to make inferences about a group of genes without investigating each one individually. Hence by using GO, each gene can be assigned its respective attributes automatically.

Terms are also separated into three categories/ontologies: Biological Process, Molecular Function and Cellular Component.

Biological Process describes biological phenomena such as a series of commonly known biological events that affects the state of an organism. Examples of biological process include cell cycle, replication of DNA etc.

Molecular Function defines the activities that take place at molecular level. It also defines the function that is carried out by a gene product. Examples of molecular function include retinoic acid receptor, glycine dehydrogenase, amino methyl transferace etc.

Cellular Component describes the location in a cell where a gene acts, where a gene product functions takes place. Examples of cellular components include nuclear inner membrane, ubiquitin ligase complex, integral membrane protein etc.

This chapter discusses in detail about the algorithms implemented in this thesis work and also gene ontology which is used to analyze gene enrichment.

CHAPTER III

RELATED RESEARCH

This chapter discusses about some previous works related to gene clustering, k-means clustering with hierarchical clustering and use of maximum-flow minimum-cut algorithm for web community discovery. This chapter contains an overview of the different algorithms previously used for gene clustering. This way the problem domain and existing solutions have been introduced in this chapter.

3.1 Research Works Related to Gene Clustering

In microarray data analysis, clustering genes to find out the biologically relevant groups based on their expression profiles is one of the basic techniques. Similarity in gene expression profiles indicates similarity in their gene functionalities also [14]. Hence the problem of grouping the genes with similar functionality that participates in the same biological process can be mapped as a clustering problem that clusters the genes based on their expression profiles [14].

So far many algorithms have been implemented for clustering gene expression data. These algorithms include hierarchical clustering [15–16], k-means clustering [1], self-organizing maps [17 – 19], support vector machines [20], Bayesian networks [21],

fuzzy logic approach [22]. Beside these algorithms, some algorithms use other genomic information along with gene expression data in order to improve clustering efficiency. These algorithms include [22] that use gene ontology data with gene expression data and [24–26] that clusters genes by using information of upstream regions of the coding sequences with gene expression profiles to get more biologically relevant clusters.

3.2 Research Works Related to Hierarchical Clustering Combined with k-means

Traditional clustering algorithms such as k-means and hierarchical clustering algorithms have already been implemented for gene clustering [1, 15–16]. As discussed in chapter 2, both k-means and hierarchical clustering method suffer from some limitation. Moreover, these algorithms are computationally expensive which impede the wide use of these algorithms in gene expression data analysis [27–29]. To overcome these limitations, a combined hierarchical k-means clustering method has been proposed in [30] which firstly applies k-means algorithm in each cluster to determine k cluster and then feed those clusters to hierarchical clustering technique to shorten merging clusters time while generating a tree-like dendrogram. But still this algorithm suffers from limitation of determining the initial value for k.

3.3 Research Works Related to Graph Clustering Using Maximum-Flow Minimum-Cut Algorithm

After calculating the correlation coefficient for all genes, a weighted graph can be created where each gene can be represented as node in the graph and at this stage clustering these genes can be mapped as a graph clustering problem. In recent research

works, solutions to the graph clustering problem have been formalized by modeling the clustering problem into a maximum-flow minimum-cut problem of the underlying graph. This approach has been used in problems like web community discovery [31–32], image segmentation etc. In [33] the authors used this approach to produce clusters and it has been shown that this approach works remarkably in practice [33]. However, in spite of wide applications of this algorithm, as the algorithm requires processing of the entire graph, if changes happen in graph structure during run time, using this algorithm becomes infeasible for dynamic graph [34]. Note that, we are using static graph in our research work, so we are not considering this limitation in this case.

CHAPTER IV

MATERIALS AND METHODS

The goal of this research work is to cluster genes where genes with similar expression level will remain in same cluster and compare their change in molecular functions for normal and cancer samples. We are using microarray data and two different approaches namely k-means clustering combined with hierarchical clustering and Ford Fulkerson algorithm for graph clustering. For determining the differentially expressed genes, we performed t-test, Bonferroni correction and calculated the value of fold change of genes in the whole dataset. This chapter discusses about the dataset used for this purpose followed by the whole process that has been carried out.

4.1 Dataset

Lung cancer is one of the leading causes of death caused by cancer worldwide [35–36]. Adenocarcinoma is the most frequent type of non-small-cell lung cancers (NSCLC) and it accounts for more than 50% of NSCLC and the percentage is increasing [37]. Recent studies have shown that activation of the EGFR, KRAS and ALK genes defines 3 different pathways which are responsible for a considerable fraction (30%–60%) of development of lung adenocarcinoma [38–42]. The remaining lung

adenocarcinomas i.e., those without EGFR, KRAS, and ALK mutations (also designated as ~~"t~~riple-negative adenocarcinomas"), develop with mutations of several other genes such as HER2, BRAF etc. However, these are known to be mutated also mutually exclusively with the EGFR, KRAS, and ALK genes though their frequencies of mutations are very low (<5%) [38–41].

The dataset used in this research work contains expression profiles for 246 samples where 20 samples belong to normal tissue. Out of 226 lung adenocarcinomas samples 127 are with EGFR mutation, 20 with KRAS mutation, 11 with EML4-ALK fusion and 68 samples are with triple negative cases. Platform used for this dataset is GPL570 [HG-U133_Plus_2] Affymetrix Human Genome U133 Plus 2.0 Array. This dataset was collected from GEO database (accession number GSE31210).

Typically expression data are analyzed in matrix form where each row represents a gene and each column represents a sample. In this study, the dataset contains 54675 genes and 40 samples which include 20 samples from normal tissue and 20 samples from KRAS positive tissues. We represent the data matrix by the symbol X and denote the data as shown in Figure 4.1. In this Figure, for example x_{22} represents the expression value of gene x_2 for Sample 2.

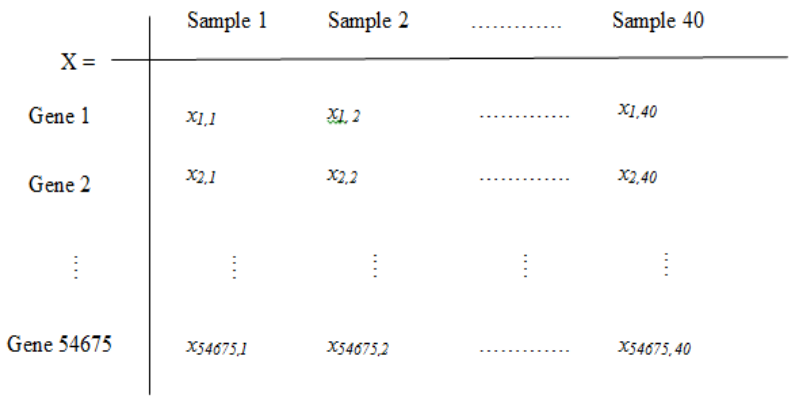

Figure 4.1: Matrix representation of the dataset

An overview of the final dataset is given in table 5.1.

4.2 Finding the Differentially Expressed Genes

To determine the differentially expressed genes, we performed t-test and Bonferroni correction followed by the calculation of the value of fold change of the genes. Brief descriptions of t-test, Bonferroni correction and fold change are given below.

4.2.1 T-Test

A t-test is a statistical hypothesis test which is used to determine if data from two sets are significantly different from each other. This test is most commonly applied to the test statistic which follows a normal distribution and the value of a scaling term in the test statistic is known.

Generally there are two types of t-test: unpaired and paired t-test. Unpaired t-test is used for the two datasets to be compared where the members of the datasets are randomly selected or otherwise not related. On the other hand, paired t-test is used for the

two datasets to be compared where the members of the datasets are related to each other and the second datasets contains the same members as the first one.

In this study, as we are comparing the value of the same gene for both normal tissue dataset and KRAS positive dataset, we used a paired t-test. Given two paired sets *X* and *Y* of *n* measured values, the paired t-test determines whether they differ from each other in a significant way under the assumptions that the paired differences are independent and identically normally distributed. The *t* statistic to test whether the means are different can be calculated as follows:

$$t = \frac{\bar{X} - \bar{Y}}{S_{XY} \cdot \sqrt{\frac{2}{n}}} \tag{4.1}$$

where

$$S_{XY} = \sqrt{\frac{1}{2}({S_X}^2 + {S_Y}^2)} \tag{4.2}$$

Here S_{XY} is the grand standard deviation. The denominator of *t* in Equation (4.1) is the standard error of the difference between two means. $\bar{X}$ and $\bar{Y}$ represent the mean values of dataset *X* and *Y* respectively, S_X and S_Y represent the standard deviation for dataset *X* and *Y* respectively and *n* is the size of the dataset.

4.2.2 Bonferroni Correction

The Bonferroni correction is a simple as well as conservative statistical method used to make adjustment to p-values when several dependent or independent statistical

tests are being performed simultaneously on a single data set. This correction aims to reduce the chances of obtaining false-positive results when multiple pairwise tests are performed on a single set of data. In order to perform a Bonferroni correction, we need to divide the critical p-value (α) by the number of comparisons being made. For example, if 10 hypotheses are being tested, the new critical p-value would be $\frac{\alpha}{10}$. The statistical power of the study is then calculated based on this modified p-value.

Let's consider a researcher is testing 20 hypotheses simultaneously, with a critical p-value of 0.05. In this case, if P denotes the probability, then the following would be true:

$$\text{P (at least one significant result)} = 1 - \text{P (no significant results)}$$

$$= 1 - (1\text{-}0.05)^{20}$$

$$= 0.64$$

This example shows that, performing 20 tests on a data set yields a 64% chance of identifying at least one significant result, even if all of the tests are actually not significant. It means while a given α may be appropriate for each individual comparison, it may not be appropriate for the set of all comparisons.

In short, Bonferroni correction tries to mitigate the risk of producing erroneous false-positive conclusions when testing multiple hypotheses on a single set of data and an appropriate use of this correction can ensure the integrity of studies in which a large number of significance tests are used.

In this study, after performing Bonferroni correction, we selected the genes as the most differentially express which have p-values ≤ 0.05.

4.2.3 Fold Change

Fold change represents a measure of how much a quantity changes going from its initial stage to a final stage. For example, if a variable has an initial value of 30 and a final value of 60, it means there is a fold change of 2, in other words, a 2-fold increase. As another example, a change from 80 to 20 would be a fold change of 0.25. Fold change is calculated simply as the ratio of the final value to the initial value. For example, if the initial value is A and final value is B, the fold change is $\frac{B}{A}$. In some cases, a fold-change value that is less than 1 can be replaced by the negative of its inverse, such as a change from 80 to 20 would be a fold change of -4, in other words, a four-fold decrease.

In this study, we considered only those genes where the value of fold change (increase or decrease) is significant. In the final dataset, we put the genes where $|log_2 f_i| \geq 1$, where f is the value of fold change for gene x_i and x is the set of genes.

Beside these preprocessing, we considered only those genes that are associated with molecular functions according to the Gene Ontology (GO). Figure 4.2 shows the flow diagram of the data preprocessing

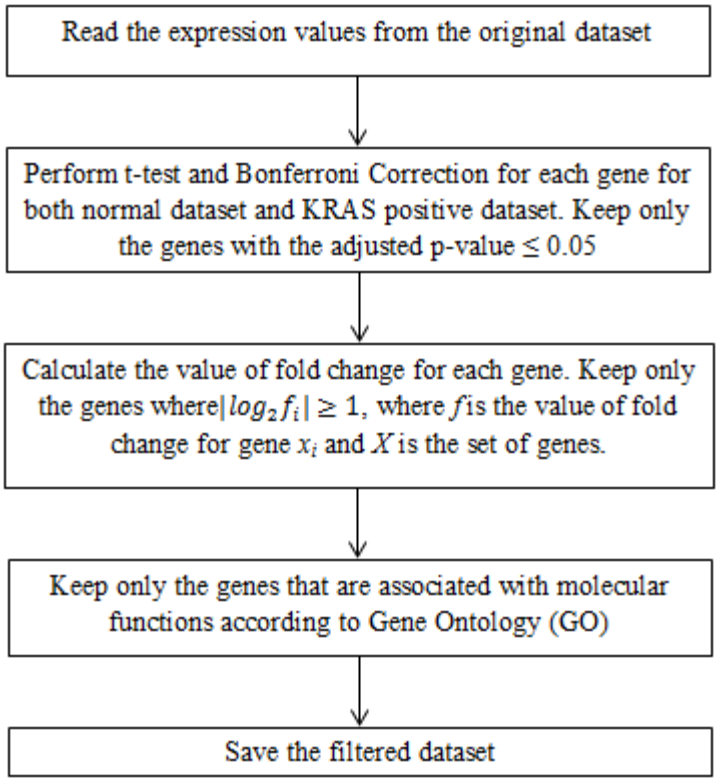

Figure 4.2: Flow diagram of data preprocessing

4.3 Methods

In this study, we investigated two approaches for grouping genes with similar expression profiles:

1. K-means clustering combined with hierarchical clustering
2. Ford-Fulkerson algorithm that uses maximum-flow minimum-cut algorithm.

We developed a tool written in Java which is used as the platform for these two approaches. Figure 4.2 shows the GUI of the tool developed in this research work.

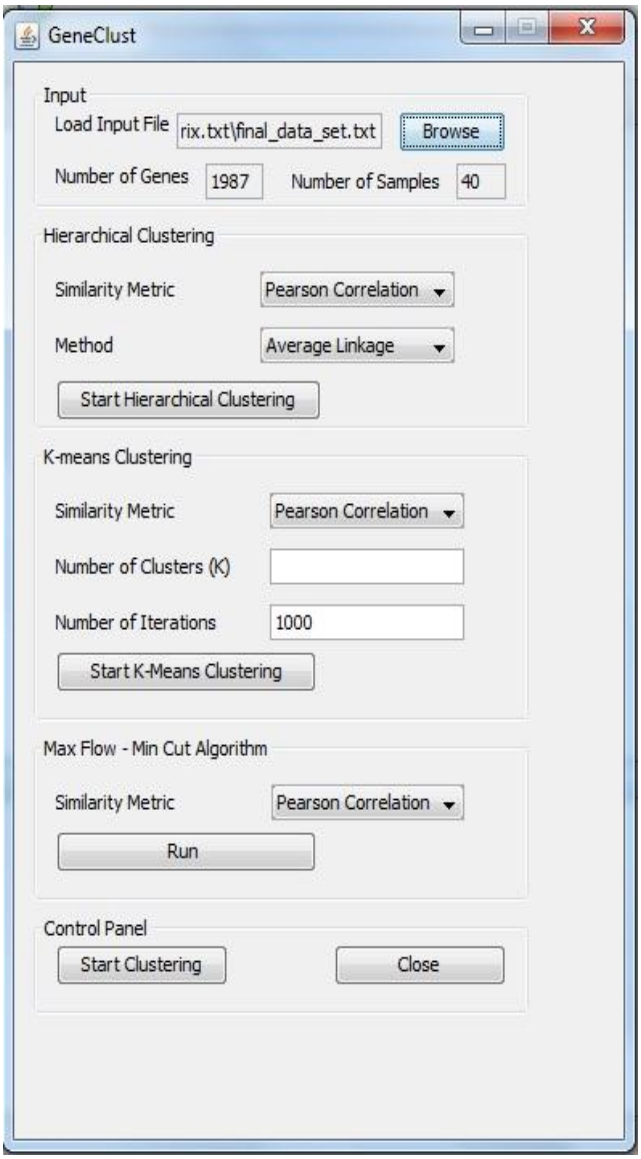

Figure 4.3: GUI of the tool developed in this research work.

4.3.1 K-means Clustering Combined with Hierarchical Clustering

As discussed in chapter 2, k-means clustering method produces tighter cluster than hierarchical clustering and also this algorithm is computationally faster than hierarchical clustering. But the performance of k-means clustering largely depends on the initial selection of the number of clusters. On the other hand, hierarchal clustering

produces a complete hierarchy of clusters which makes it easy to understand how the objects group while clustering. So to overcome the limitation of k-means clustering, here we used a combined approach to decide the number of clusters for the k-means clustering from the output of hierarchical clustering. Flow chart of this approach is shown in Figure 4.3.

4.3.2 Ford Fulkerson Algorithm for Graph Clustering

From the dataset, a weighted graph can be created where each node is represented by a gene and the weight (capacity) between two nodes is the value of Pearson correlation coefficient between the corresponding genes. After creating the weighted graph, Ford Fulkerson algorithm can be applied to get the clusters where genes with similar expression profile will group together.

To apply Ford Fulkerson algorithm, we need to specify a source node and a sink node. For this purpose we used Dijkstra's algorithm for each pair of nodes and then the two nodes with maximum shortest distance were selected as source and sink nodes. Note that, as Dijkstra's algorithm deals with distance, we used Pearson correlation distance while applying Dijkstra's algorithm. After getting the minimum cut graph from Ford Fulkerson algorithm in the form of two disjoint sets, we recursively apply Ford Fulkerson algorithm again to each set until the cardinality of the disjoint set is less than or equal to 2. In this approach, clustering is done in a top down fashion. Figure 4.4 shows the flow diagram of the clustering process.

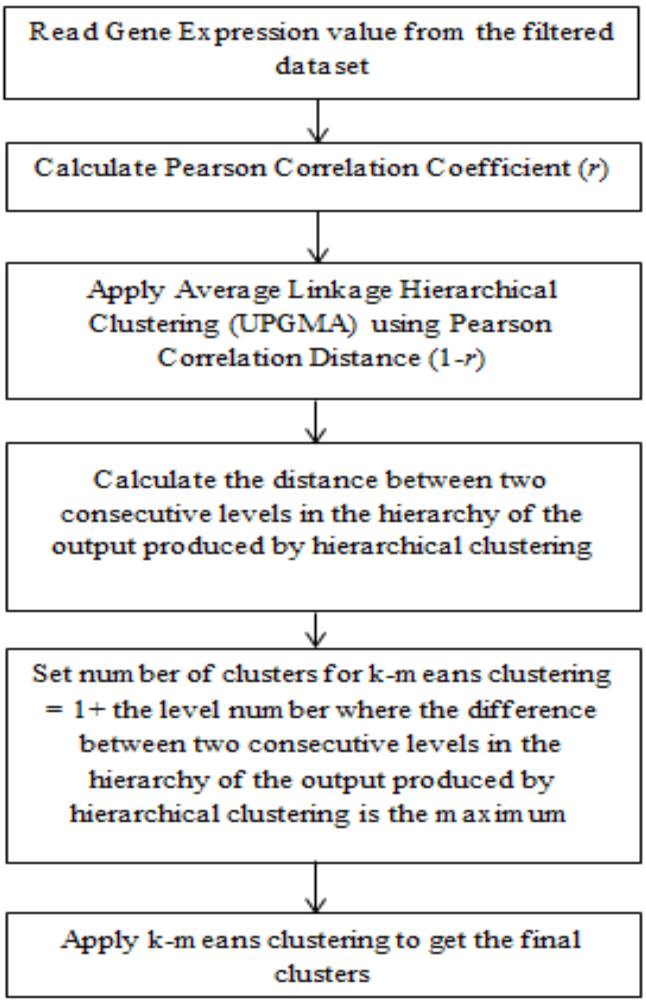

Figure 4.4: Flow diagram of K-means clustering combined with hierarchical clustering

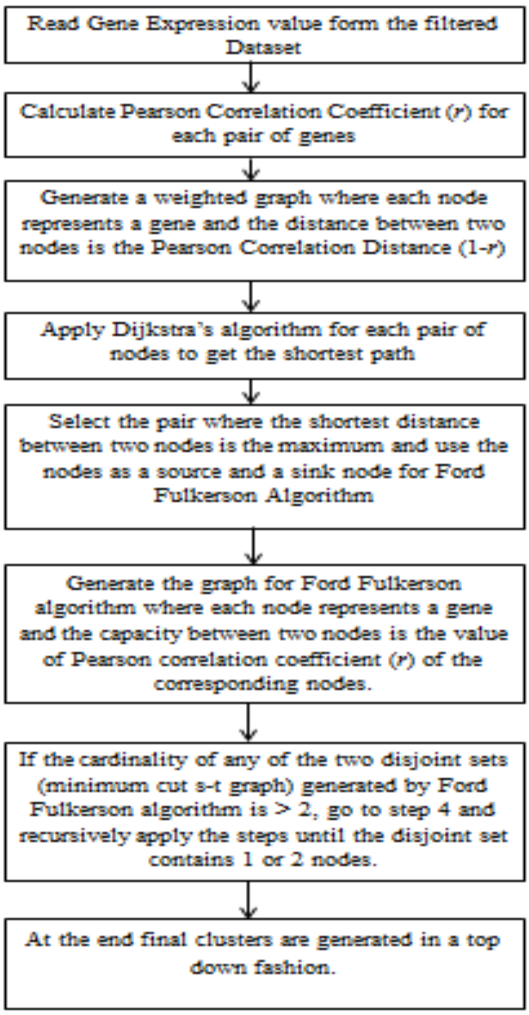

Figure 4.5: Work flow diagram of applying Ford Fulkerson algorithm for clustering

4.4 Comparing Molecular Functions of the Genes

We explored the molecular functions captured in each cluster of genes using Gene Ontology (GO). For each cluster, the molecular functions obtained from normal lung tissues are compared to the ones from the KRAS positive tissues. The result illustrates the change in molecular functions which is the underlying reason for cancer formation and development.

CHAPTER V

RESULTS AND DISCUSSION

In this study, we used two approaches for clustering the genes based on their expression levels. These approaches are: k-means clustering combined with hierarchical clustering and a maximum-flow minimum-cut based approach. In this chapter the results obtained by these two approaches are presented and a comparative study of the molecular functions of the genes in the clusters is done using gene ontology annotation. Section 5.1 briefly shows dataset containing highly expressed genes we obtained after preprocessing using t-test, Bonferroni correction and calculating the value of fold change. Section 5.2 presents the results obtained by using the k-means clustering combined with hierarchical clustering approach and followed by the result obtained from the maximum-flow minimum-cut approach which is presented in section 5.3. The molecular function of the genes captured in each cluster for cancer and normal data are explored using gene ontology annotation and presented in section 5.4.

5.1 Preprocessing of Dataset

Initially the dataset contained 54675 genes and 40 samples (20 for normal tissues and 20 for KRAS positive tissues). We determined the highly expressed genes using the t-test followed by Bonferroni correction and calculating the value of fold change. After

performing t-test we obtained 21880 genes which had significant p value (≤ 0.05). We performed Bonferroni correction on these genes and found 1988 genes which had a significant adjusted p value (≤ 0.05). As the total number of genes was still too high, we calculated the value of fold change and got 1005 genes which had $|log_2 f_i| \geq 1$, where f is the value of fold change for gene x_i and X is the set of genes. We then performed another step of filtering to keep only those genes that have Gene Ontology (GO) terms and responsible for molecular functions. Finally we came up with 464 genes in the dataset. The final dataset is given partially in Table 5.1 and the complete dataset is available in [43].

Table 5.1: A brief overview of the final dataset

Affymatrix ID	Gene Symbol	Samples				
		GSM773551	GSM773552		GSM773784	GSM773785
1555579_s_at	PTPRM	3441.222	2205.079		3569.134	5426.586
211986_at	AHNAK	4395.679	3074.898		7080.395	8986.732
222392_x_at	PERP	21707.73	11773.66		11350.53	9255.438
236715_x_at	UACA	1303.009	685.552		1867.759	2359.538
244704_at	NFYB	124.0794	103.3855		277.4942	303.6087
…	…	…	…		…	…
211237_s_at	FGFR4	22.41334	6.840419		11.07046	134.4622
203980_at	FABP4	257.254	28.62955		920.4353	5008.182
207302_at	SGCG	47.08894	4.437844		9.613587	368.0018
210081_at	AGER	241.6255	26.12709		2001.279	4878.16
217046_s_at	AGER	132.4155	21.06835		1016.052	2485.816

5.2 Clustering Results from K-means Combined with Hierarchical Clustering

As discussed earlier, k-means clustering algorithm requires the initial number of clusters before starting the clustering and the performance as well as the subsequent

interpretation of the clusters largely depends on this number. In this approach we tried to overcome the limitation of k-means clustering by getting the initial number of clusters from the result of hierarchical clustering. Figure 5.1 shows the hierarchical clustering of normal tissue dataset.

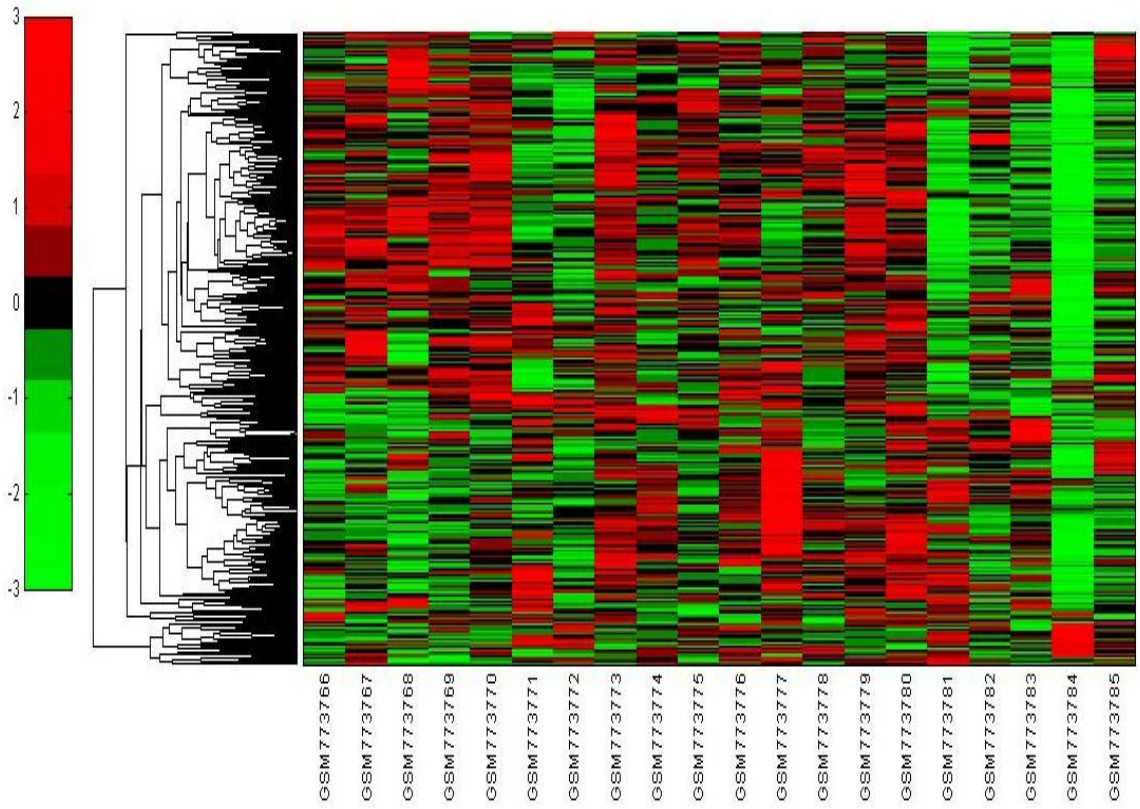

Figure 5.1: Hierarchical Clustering of normal tissue dataset

There are 463 nodes in this tree generated from the hierarchical clustering. To decide the number of clusters from the output of hierarchical clustering we used a bar graph to show the difference of height between two consecutive nodes and it is shown in Figure 5.2.

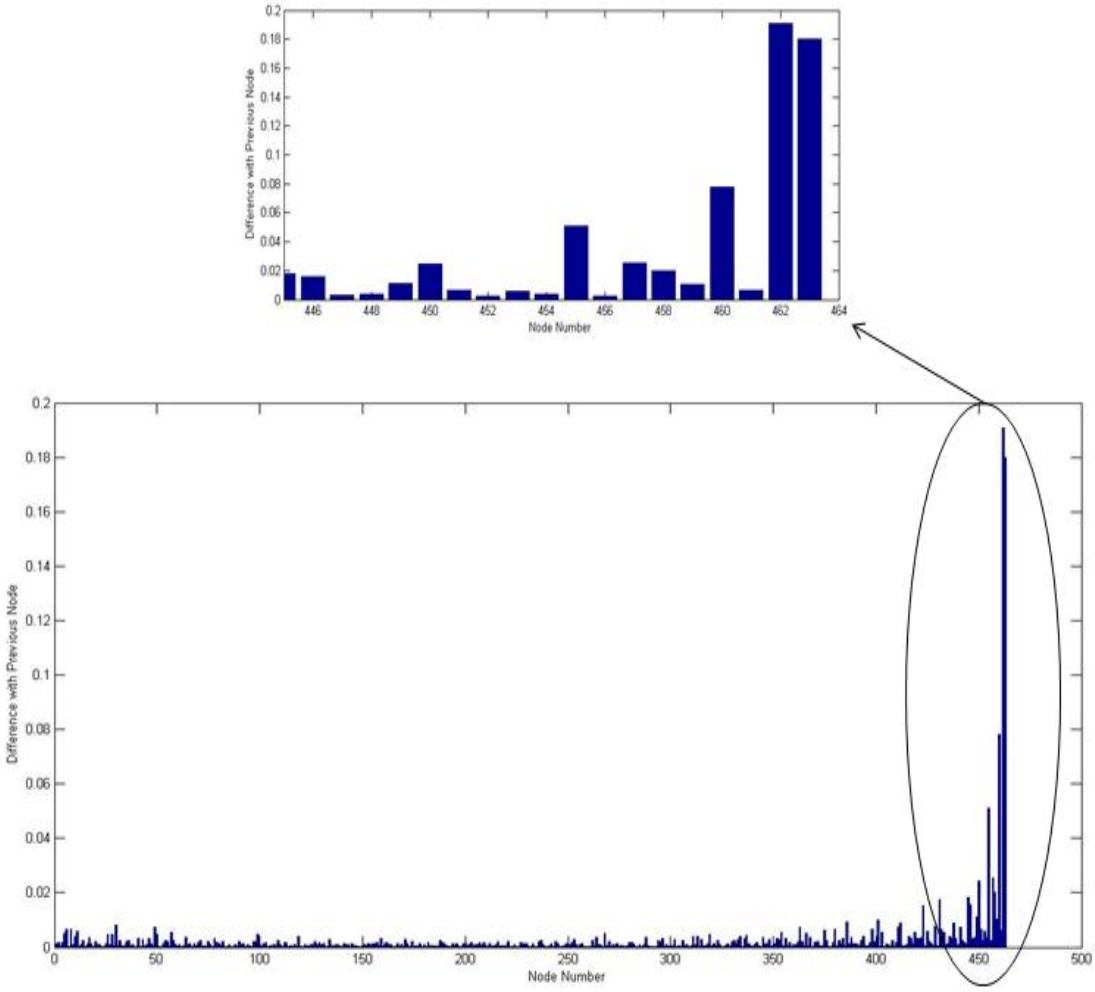

Figure 5.2: Bar graph of the difference of height between two consecutive nodes in the tree generated from the hierarchical clustering of normal tissue dataset.

From Figure 5.2 we can see that the difference is the maximum for node 461 and node 462. As there are total 463 nodes in the tree, node 461 is in level 3 from the top. So according to the approach we are discussing here, the total number of clusters for k-means clustering should be 4.

List of the genes for cluster 1 for normal tissue dataset is given in Table 5.2. List of the genes for cluster 2, 3, 4 are given in the appendix (See Table App 1.1, App 1.2 and App 1.3). Note that, some genes may have same gene symbols though they have different Affymatrix IDs. So there are some duplicate gene symbols in the tables.

Table 5.2: List of the genes contained in Cluster 1 for the normal tissue dataset

PERP	TSSC1	ADCY4	SRSF3	P4HB	COPZ1
ATP6V0A2	PCM1	PCBD2	UGGT1	RHOJ	MYO7A
IDE	KGFLP2	SSR4	CANT1	KIF2A	ATP2A2
EFEMP1	SYNPO	PAK1	COPB1	IDS	CYCS
SEC24A	IFT57	RHOJ	ID4	ECT2	RHOJ
RALGPS2	SFN	RBPMS	CDC25A	SFN	RALGPS2
FERMT1					

Similarly we can perform the combined clustering on KRAS positive dataset. Figure 5.3 shows the hierarchical clustering of KRAS positive dataset.

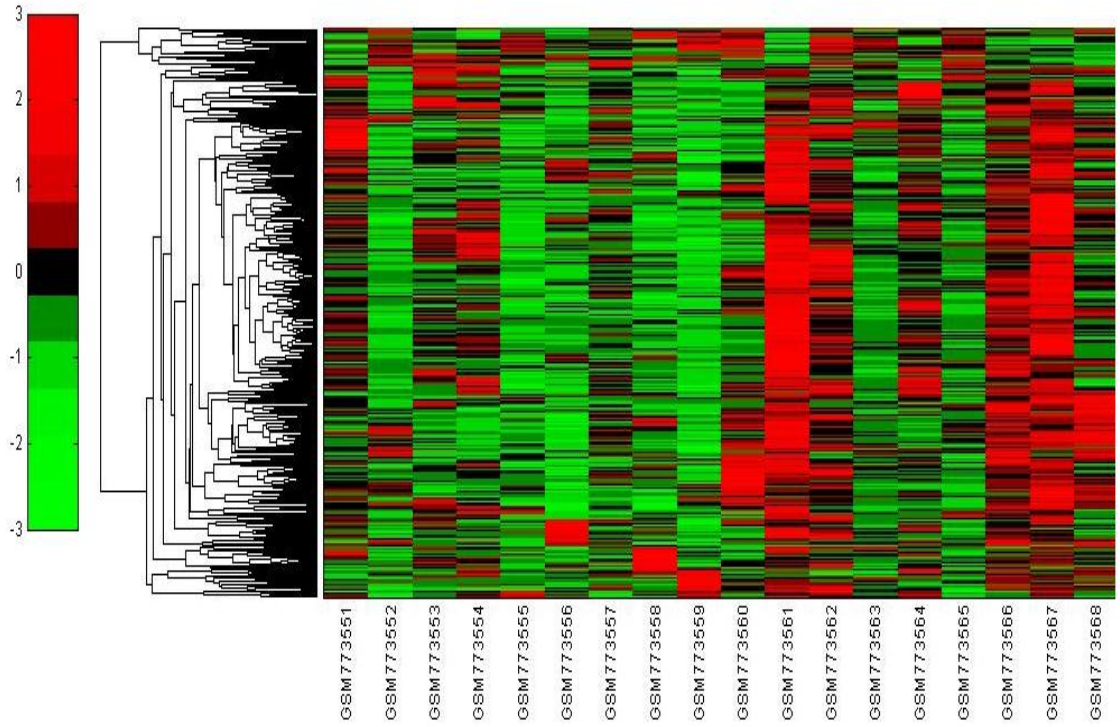

Figure 5.3: Hierarchical Clustering of KRAS positive dataset

There are 463 nodes in this tree generated from the hierarchical clustering. To decide the number of clusters from the output of hierarchical clustering we used a bar graph to show the difference of height between two consecutive nodes and it is shown in Figure 5.4.

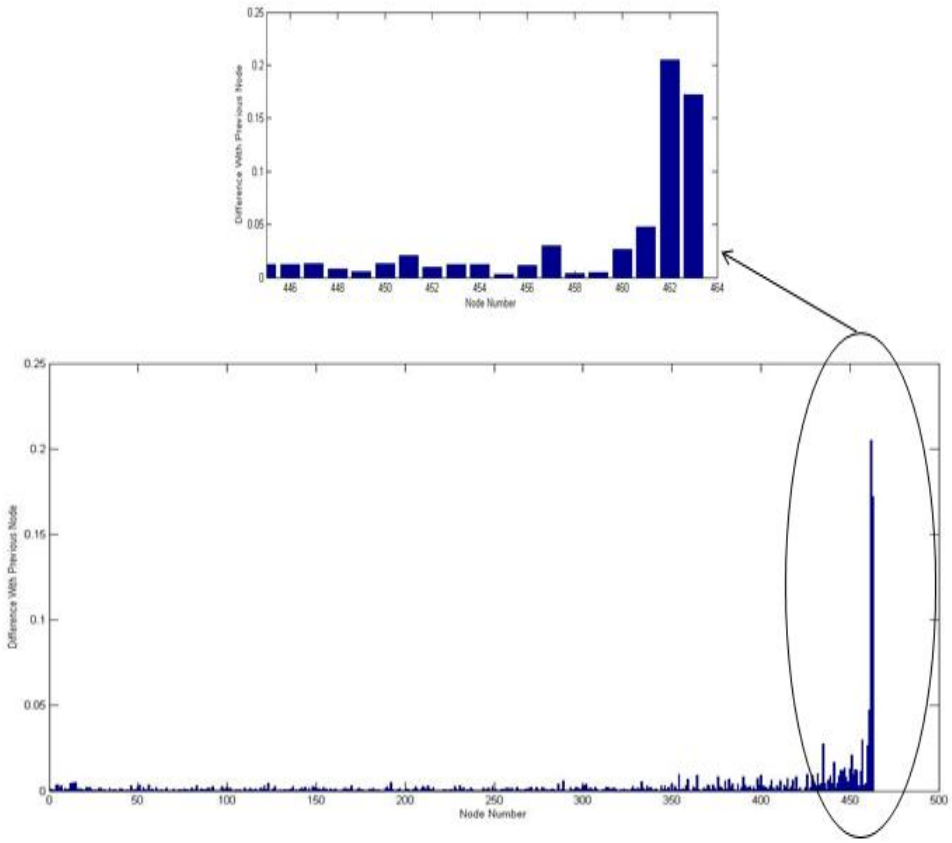

Figure 5.4: Bar graph of the difference of height between two consecutive nodes in the tree generated from the hierarchical clustering of KRAS positive dataset.

From Figure 5.4 we can see that the difference is the maximum for node 461 and node 462. As there are total 463 nodes in the tree, node 461 is in level 3 from the top. So according to the approach we are discussing here, the total number of clusters for k-means clustering should be 4.

List of the genes for cluster 1 for KRAS positive dataset is given in Table 5.3. List of the genes for cluster 2, 3, 4 are given in the appendix (See Table App 1.4, App 1.5 and

App 1.6). Note that, some genes may have same gene symbols though they have different Affymatrix IDs. So there are some duplicate gene symbols in the tables.

Table 5.3: List of genes contained in Cluster 1 for the KRAS positive dataset

PERP	TSSC1	P4HB	TIMP3	FAM69A	ITPR1
GAPDH	PCBD2	UGGT1	MYO7A	IDE	CBR4
SSR4	ICAM2	AKT3	CANT1	KIF2A	ATP2A2
FANCD2	CDKN1C	PAK1	SEC24A	AKAP12	AKAP12
E2F2	SPOCK2	RUNX1	SKA3	ECT2	RALGPS2
SFN	GPR82	CDC25A	SFN	RALGPS2	FERMT1

5.3 Clustering Results from Maximum Flow Minimum Cut Approach

After calculating the Pearson Correlation Coefficient, a weighted graph can be formed where each node is represented by a gene and the weight of the edge between two genes can be represented by the corresponding Pearson correlation coefficient. As discussed in Chapter 4, maximum flow minimum cut approach can be used for graph clustering and Ford Fulkerson algorithm is commonly used in clustering web graph to discover web communities. In this research work, we used this approach for clustering the gene graph so that genes with high similarity can be grouped together. Note that, Ford Fulkerson algorithm requires source and sink node at the beginning and we used Dijkstra's algorithm to find out the source and the sink node as discussed in Chapter 4.

Figure 5.5 shows a brief overview of how the maximum flow minimum cut algorithm works to a small part of the normal tissue dataset.

Let's consider Set_1 which contains all the genes. Hence,

Step 1: Set_1 = {CYCSKIF2A, IDE, FERMT1SFN, SFN, ATP2A2, SEC24A, UGGT1, ECT2, PAK1, COPZ1, CANT1, P4HB}

Step 2: After applying the Ford-Fulkerson algorithm to Set_1, we get two disjoint sets, say $Set_{1,1}$ and $Set_{1,2.}$

$Set_{1,1}$ = {CYCSKIF2A, IDE, FERMT1SFN, SFN, ATP2A2, SEC24A, UGGT1, ECT2}

$Set_{1,2}$ = {PAK1, COPZ1, CANT1, P4HB}

Step 3: After applying the Ford-Fulkerson algorithm to Set1,1, we get two disjoint sets, say $Set_{1,1,1}$ and $Set_{1,1,2}$.

Set $_{1,1,1}$ = {CYCSKIF2A, IDE, FERMT1SFN, SFN, ATP2A2}

$Set_{1,1,2}$ = {SEC24A, UGGT1, ECT2}

In Step4, the algorithm is applied to the Set 1, 1, 1and this process is applied to all sets, until the set has a cardinality which is less than or equal to 2. In this way, the tree for clustering is formed in a top to bottom fashion.

After applying the Ford-Fulkerson algorithm on the dataset of normal tissue and KRAS positive tissues, we found that the clustering result were same as the hierarchical clustering hence it proves the correctness of this maximum flow minimum cut based approach.

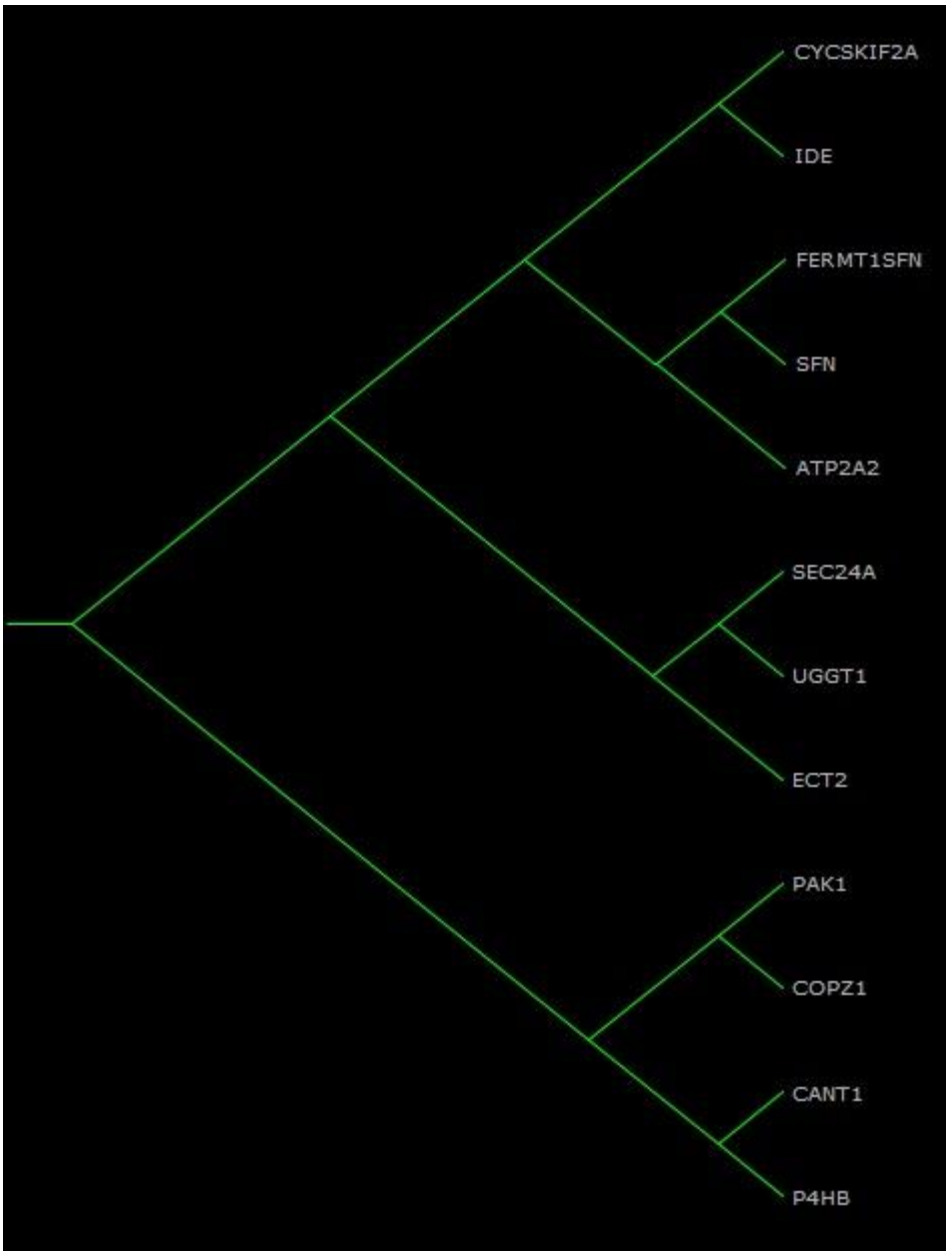

Figure 5.5: A brief overview of how the maximum flow minimum cut algorithm works.

5.4 Discussion and Analysis

In this section we explain the change molecular function of the genes captured in the clusters of both normal tissue and KRAS positive datasets using Gene Ontology (GO)

annotations. For comparing the molecular function of the clusters of normal tissue and KRAS positive tissues, we took one cluster from normal tissue dataset and one from KRAS positive dataset which have maximum number of common genes. Table 5.10 shows the clusters we have selected for comparing their molecular functions with the number of genes they have in common.

Table 5.4: List of the clusters to be compared for the change in molecular function

Clusters to compare		Number if genes in common
Normal Tissue dataset	KRAS positive dataset	
Cluster 1	Cluster 1	20
Cluster 2	Cluster 3	52
Cluster 3	Cluster 4	46
Cluster 4	Cluster 2	69

We explained the molecular functions of the genes in each cluster using GO annotations and the relationship are represented using a Directed Acyclic Graph (DAG) which is termed as GO graph in this study. To generate these graph we used a web based tool named GOEAST which stand for Gene Ontology Enrichment Analysis Software Toolkit [44]. This graph displays enriched GOIDs and their hierarchical relationships in "molecular function" GO categories. Here boxes represent GO terms, labeled by its Gene Ontology ID (GOID), term definition, p-value etc. Note that significantly enriched GO terms are marked yellow. The degree of color saturation of each node is positively correlated with the significance of enrichment of the corresponding GO term. Non-significant GO terms within the hierarchical tree are shown as white boxes. In this graph, edges stand for connections between different GO terms. Edges with red color stand for relationship between two enriched GO terms, black solid edges stand for relationship

between enriched and un-enriched terms; black dashed edges stand for relationship between two un-enriched GO terms.

5.4.1 Comparing Cluster 1 of Normal Tissue with Cluster 1 of KRAS Positive Tissues

Figure 5.6 and 5.7 shows the GO graph for the cluster 1 of normal tissue dataset and cluster 1 of KRAS positive dataset respectively.

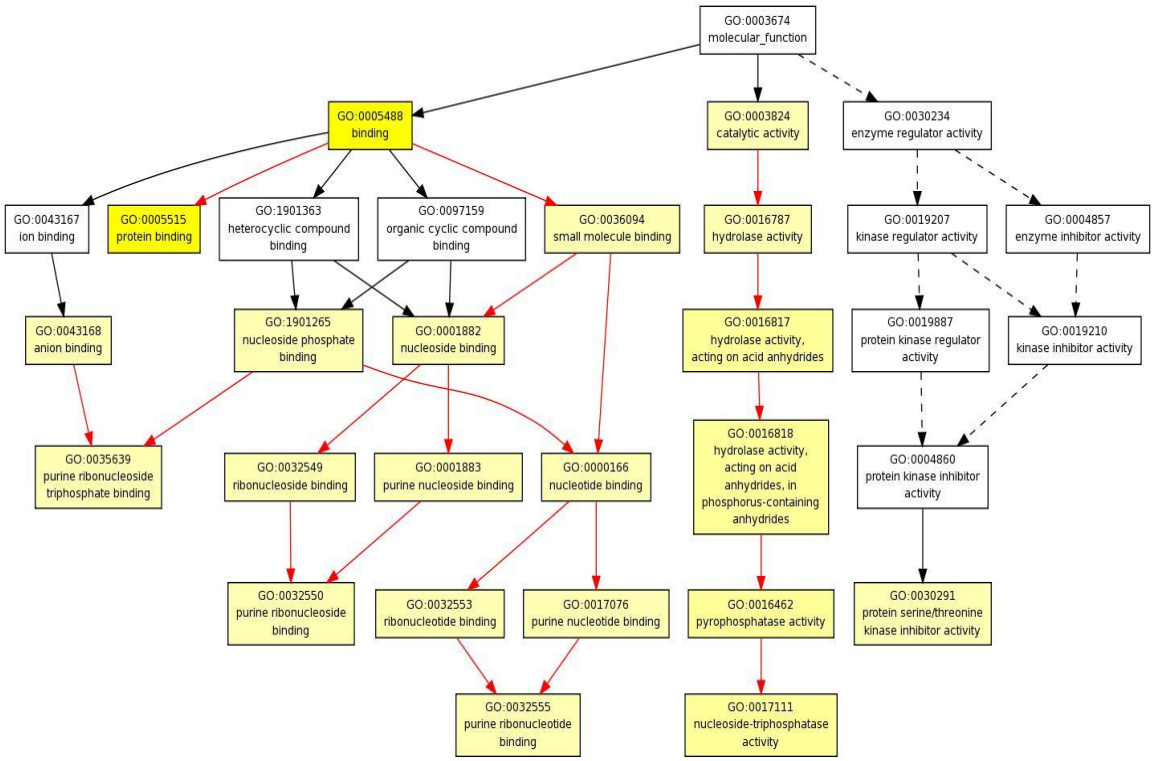

Figure 5.6: GO graph for cluster 1 of normal tissue data set.

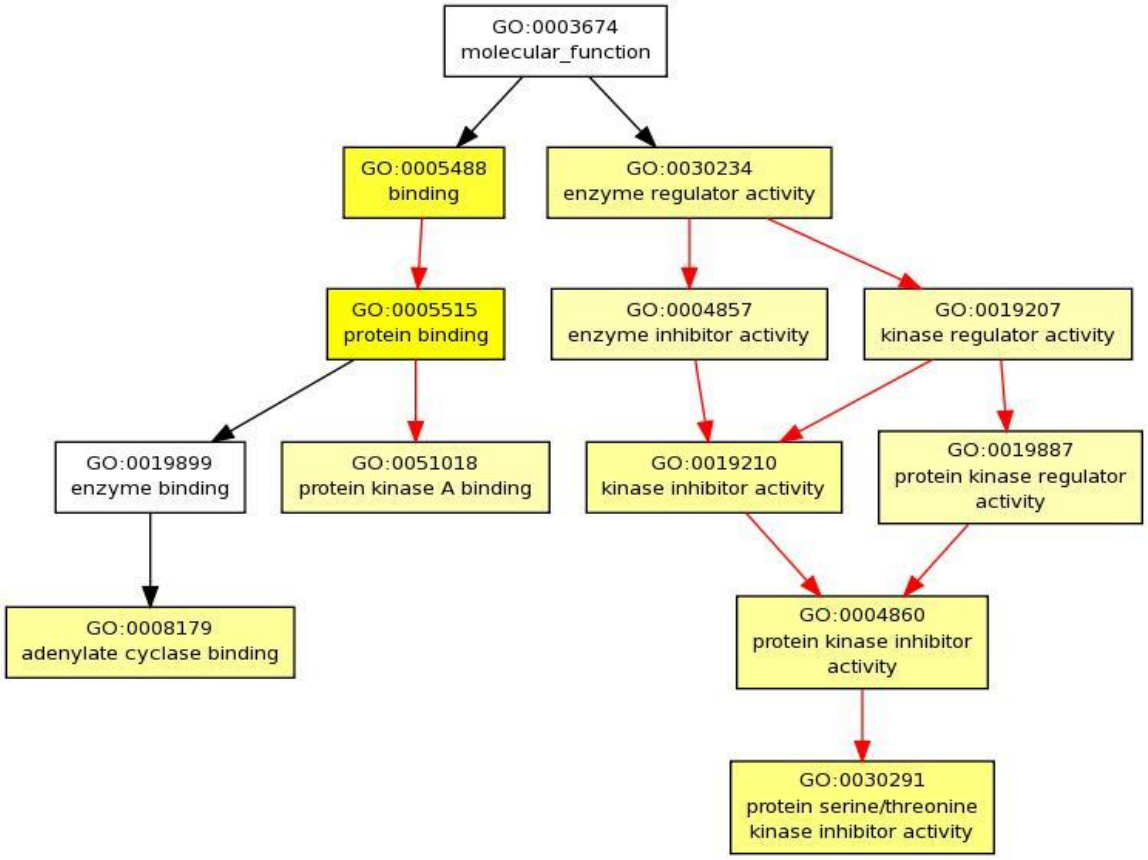

Figure 5.7: GO graph for cluster 1 of KRAS positive data set

In brief, from these two figures we see that, the significant GO terms GO: 0005488 (binding) and GO: 0005515 (protein binding) remain same in both clusters. GO terms such as GO: 0030234 (Enzyme Regulator Activity), GO: 0019207 (Kinase Regulator Activity), GO: 0019210 (Kinase Inhibitor Activity), GO: 0019887 (Protein Kinase regulator Activity) and GO: 0004860 (Protein Kinase Inhibitor Activity) which are un-enriched in normal tissue, become highly enriched in the KRAS positive tissues.

For better comparing the enrichment status of the two clusters, we used Multi-GOEAST which is an advanced version of GOEAST and it is helpful to identify the

hidden correlation between the two clusters. Figure 5.8 shows the comparative GO graph of the clusters discussed above.

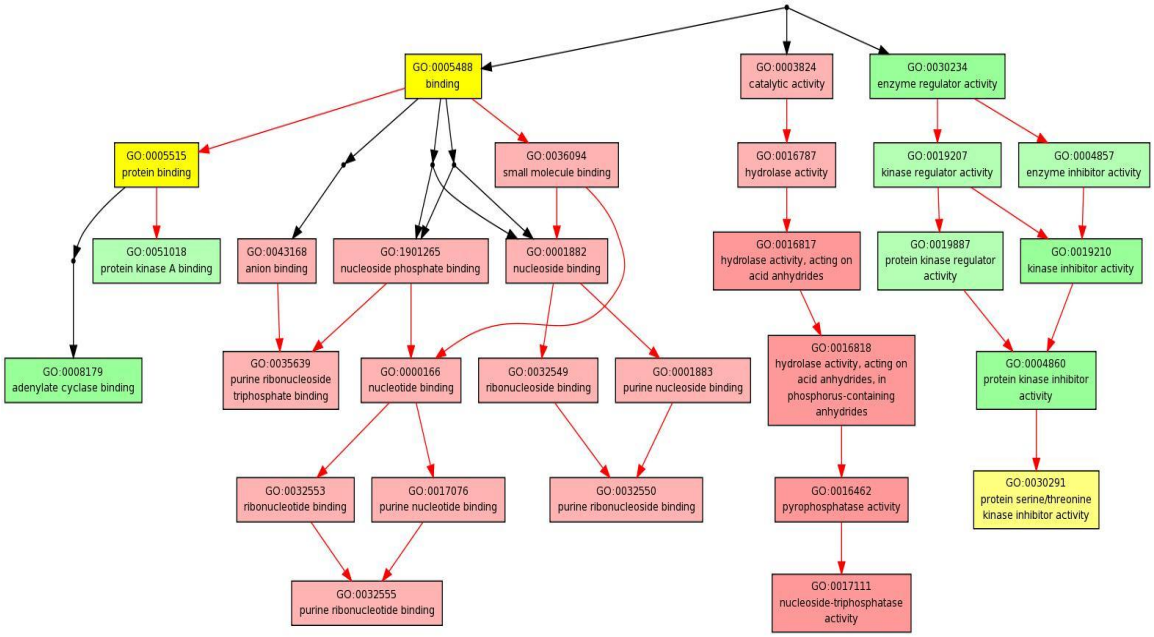

Figure 5.8 Comparative GO graph for comparing GO enrichment status of Cluster 1 of normal tissue dataset and Cluster 1 of KRAS positive dataset.

In the comparative GO graph, significantly enriched GO terms in both clusters are marked yellow, light yellow color indicates the GO terms which are enriched in both clusters. Nodes marked with coral pink indicate the GO terms which are enriched in normal tissue dataset but not in KRAS positive dataset. In addition to that, nodes with green color represent the GO terms which are un-enriched in normal tissue but enriched in KRAS positive tissues. Note that, the degree of color saturation of each node is positively correlated with the significance of enrichment of the corresponding GO term.

Table 5.5 lists the genes associated with the GO terms which are enriched in the cluster 1 of KRAS positive tissue dataset but not enriched in the cluster 1 of normal tissue

dataset and these GO terms which are marked with green color in the comparative GO graph shown in Figure 5.8. These are responsible for the change in the molecular activity of the genes that causes the development of lung cancer.

Table 5.5: GO Terms and pathways which are enriched in molecular functions of the genes of Cluster1 of KRAS positive tissue but un-enriched in the genes of Cluster1 of Normal tissue dataset

GO ID	GO Term	Associated Genes	Pathway
GO:0030234	Enzyme Regulator Activity	TIMP3	Matrix_Metalloproteinases
		CDKN1C	G1_to_S_cell_cycle_Reactome
		PAK1	Integrin mediated_cell_adhesion_KEGG
		ECT2	-----
		RALGPS2	-----
		SFN	Calcium_regulation_in_cardiac_cells
			Smooth_muscle_contraction
GO:0019207	Kinase Regulator Activity	CDKN1C	G1_to_S_cell_cycle_Reactome
		SFN	Calcium_regulation_in_cardiac_cells
			Smooth_muscle_contraction
GO:0004857	Enzyme Inhibitor Activity	TIMP3	Matrix_Metalloproteinases
		CDKN1C	G1_to_S_cell_cycle_Reactome
		SFN	Calcium_regulation_in_cardiac_cells
			Smooth_muscle_contraction
GO:0019887	Protein Kinase Regulator Activity	CDKN1C	G1_to_S_cell_cycle_Reactome
		SFN	Calcium_regulation_in_cardiac_cells
			Smooth_muscle_contraction
GO:0019210	Kinase Inhibitor Activity	CDKN1C	G1_to_S_cell_cycle_Reactome
		SFN	Calcium_regulation_in_cardiac_cells
			Smooth_muscle_contraction

GO:0004860	Protein Kinase Inhibitor Activity	CDKN1C	G1_to_S_cell_cycle_Reactome
		SFN	Calcium_regulation_in_cardiac_cells
			Smooth_muscle_contraction
GO:0051018	Protein Kinase A Binding	AKAP12	G_Protein_Signaling
GO:0008179	Adenylate Cyclase Binding	AKAP12	G_Protein_Signaling

Similarly we can generate and compare the GO enrichment graph for the rest of the clusters listed in Table 5.4 which are given in the Appendix.

This chapter discusses about the result of applying both of the clustering approaches to the dataset. Additionally, this chapter analyzes the change in GO enrichment of molecular functions of the genes captured in the clusters for both normal tissue and KRAS positive tissues.

CHAPTER VI

CONCLUSION

The aim of the study is to group biologically relevant genes by using different approaches of gene clustering. In the first approach a combined algorithm is used to cluster genes using k-means clustering algorithm where the initial number of clusters is decided from the output of hierarchical clustering. This approach overcomes the limitation of both k-means clustering and hierarchical clustering discussed in previous chapters. In the second approach, we used maximum-flow minimum-cut based algorithm Ford-Fulkerson algorithm which is a commonly used algorithm for discovering web communities by clustering web graphs. This approach produced similar result as the hierarchical clustering hence proves the correctness of this approach in gene clustering.

In this study we examined 40 samples and 464 genes from the dataset of Adenocarcinoma which is the most frequent type of non-small-cell lung cancers. Out of the 40 samples, 20 were from normal tissue and 20 were from KRAS positive tissues. We applied t-test, Bonferroni correction and Fold Change to find the significantly differentially expressed genes and included only these genes in the final dataset.

After applying the clustering algorithms we obtained 4 clusters for both normal tissue dataset and KRAS positive dataset. Hereafter, we examined the genes contained in

each cluster with respect to their molecular functions based on Gene Ontology (GO) annotation to see what are the changes in the enrichment of the molecular functions of the genes took place from normal tissues to KRAS positive tissues.

The k-means clustering algorithm combined with hierarchical clustering takes the advantage of hierarchical clustering to get a complete hierarchy of clusters and using this information it decides the initial number of clusters to be used in k-means clustering which produces a tighter cluster than hierarchical clustering. This way it overcomes the limitation of k-means clustering. In the second approach we found that the maximum-flow minimum-cut based gene clustering produces same result as hierarchical clustering hence it proves the correctness of this approach. Therefore, we propose that both of these approaches can be used for clustering microarray data.

REFERENCES

[1] S. Tavazoie, J.D. Hughes, M.J. Campbell, R.J. Cho, and G.M. Church (1999), –Systematic determination of genetic network architecture", Nature Genetics 22:3, pp. 281–285.

[2] F Luo, K Tang, and L Khan (2003), –Hierarchical clustering of gene expression data", Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering, pp. 328 -335.

[3] G. W. Flake, R. E. Tarjan and K. Tsioutsiouliklis (2004), –Graph clustering and minimum cut trees", Internet Mathematics (1: 3), pp. 385-408.

[4] Z. Wu and R. Leahy (1993) –An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation." IEEE Transactions on Pattern Analysis and Machine Intelligence 15:11, pp. 1101—1113.

[5] G. W. Flake, S. Lawrence, and C. L. Giles (2000), –Efficient Identification of Web Communities." In Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining, pp. 150—160. New York, ACM Press.

[6] L. Hunter (1993), –Artificial Intelligence and Molecular Biology", Cambridge, USA: MIT Press,

[7] G.S. Michaels, D.B. Carr, M. Askenazi, S. Fuhrman, X. Wen, R. Somogyi (1998), –Cluster analysis and data visualization of large scale gene expression data", Proceedings of the Pacific Symposium on Biocomputing, pp. 3:42-53.

[8] N. Speer, C. Spieth, A. Zell (2004), –A Memetic Co-Clustering Algorithm for Gene Expression Profiles and Biological Annotation", Proceedings of the 2004 Congress on Evolutionary Computation, USA, pp. 2:1631-1638.

[9] D.W. Mount (2001), –Bioinformatics: Sequence and Genome Analysis", New York: Cold Spring Harbor Laboratory Press.

[10] J. L. Rodgers and W. A. Nicewander (1988), –Thirteen ways to look at the correlation coefficient", The American Statistician 42: 1, pp. 59–66.

[11] S. M Stigler (1989), "Francis Galton's Account of the Invention of Correlation", Statistical Science 4:2, pp. 73–79.

[12] M. M. Babu (2004), –An Introduction to Microarray Data Analysis" in –Computational Genomics: Theory and Application", pp. 225 - 249, Cambridge, UK: Laboratory of Molecular Biology.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein (2009), –Introduction to Algorithms" 3rd Edition, Cambridge, MA: MIT Press.

[14] F. Azuaje, J. Dopazo (2005), –Data Analysis and Visualization in Genomics and Proteomics", New Jersey: Wiley.

[15] M. B. Eisen, P. T. Spellman, P. O. Brown, D. Botstein (1998), –Cluster analysis and display of genome-wide expression patterns", Proceedings of the National Academy of Science, USA, pp. 95: 14863-14868.

[16] X. Wen, S. Fuhrman, G. S. Michaels, D. B. Carr, S. Smith, J. L. Barker, R. Somogyi (1998), –Large-scale temporal gene expression mapping of central nervous system development", Proceedings of National Academy of Science, USA, pp. 95: 334-339.

[17] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, T. R. Golub (1999), –Interpreting patterns of gene expression with self-organizing maps : Methods and application to hematopoietic differentiation", Proceedings of National Academy of Science, USA, pp. 96: 2907-2912.

[18] P. Törönen, M. Kolehmainen, G. Wong, E. Castren (1999), –Analysis of gene expression data using self-organizing maps" FEBS Letter 451: 2, pp. 142-146.

[19] J. He, A. H. Tan, C. L. Tan (2003), –Self-organizing Neural Networks for Efficient Clustering of Gene Expression Data", Proceedings of International Joint Conference on Neural Networks, USA, pp. 1684-1689.

[20] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T.S. Furey, M. Ares, D. Haussler (2000), –Knowledge-based analysis of microarray gene expression data by using support vector machines", Proceedings of National Academy of Science, pp. 97: 262-267.

[21] N. Friedman, M. Linial, I. Nachman, D. Peer (2000), –Using Bayesian networks to analyze gene expression data", Journal of Computational Biology, pp. 7: 601-20.

[22] P. J. Woolf, Y. Wang (2000), –A fuzzy logic approach to analyzing gene expression data", Physiol Genomics, pp. 3: 9-15.

[23] H. Wang, F. Azuaje, O. Bodenreider (2005), –An Ontology-Driven Clustering Method for Supporting Gene Expression Analysis", Proceedings of the 18th IEEE International Symposium on CBMS, pp. 389-394.

[24] I. Holmes, W. J. Bruno (2000), –Finding regulatory elements using joint likelihoods for sequence and expression profile data", Proceedings of International

Conference on Intelligent Systems for Molecular Biology, San Diego, USA, pp. 8: 202-210.

[25] Y. Barash, N. Friechman (2002), –Context-specific Bayesian clustering for gene expression data". Journal of Computational Biology, pp. 9: 169-191.

[26] J. Kasturi, R. Acharya, and M. Ramanathan (2003), –An information theoretic approach for analyzing temporal patterns of gene expression" Bioinformatics (19: 4), pp. 449-458.

[27] G. Garai and B. B. Chaudhuri (2004), "A novel genetic algorithm for automatic clustering", Pattern Recognition Letters (25: 2), pp. 173-187.

[28] K. Ushizawa, C. B. Herath, K. Kaneyama, S. Shiojima, A. Hirasawa, T. Takahashi, K. Imai, K. Ochiai, T. Tokunaga, Y. Tsunoda, G. Tsujimoto, K. Hashizume (2004), "cDNA microarray analysis of bovine embryo gene expression profiles during the pre-implantation period," Reproductive Biology and Endocrinology (2:77).

[29] N. Bolshakova, F. Azuaje, and P. Cunningham (2005), "An integrated tool for microarray data clustering and cluster validity assessment," Bioinformatics (2: 1), pp. 451-455.

[30] T. Tsai, Y. Chen, C. Lin, R. Chen, S. Li, H. Chen (2005), –A combined K-means and hierarchical clustering method for improving the clustering efficiency of microarray", International Symposium on Intelligent Signal Processing and Communication Systems, Hong Kong, pp. 405 – 408.

[31] G. W. Flake, S. Lawrence, and C. L. Giles (2000), –Efficient identification of web communities", Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 150–160.

[32] Z. Wu and R. Leahy (1993), –An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation", IEEE Transaction on Pattern Analysis and Machicne (15:11), pp. 1101–1113.

[33] R. Kellogg, A. Heath, and L. Kavraki (2004), –Clustering Metabolic Networks Using Minimum Cut Trees", http://cohesion.rice.edu/engineering/computerscience/emplibrary/AHeath.ppt, last accessed on 5/13/2013.

[34] B. Saha and P. Mitra (2006), "Fast Incremental Minimum-Cut Based Algorithm for Graph Clustering", Proceedings of 6th IEEE International Conference on Data Mining Workshops, pp. 207-211.

[35] J. Ferlay, HR Shin, F. Bray, D. Forman, C. Mathers, DM Parkin (2010), –Estimates of worldwide burden of cancer in 2008:GLOBOCAN2008", International Jornal of Cancer, pp. 127:2893–917.

[36] DM Parkin, F. Bray, J. Ferlay, P. Pisani (2005), –Global cancer statistics 2002", CA Cancer J Clin, pp.55:74–108.

[37] H. Okayama, et al. (2012), –Identification of genes upregulated in ALK-positive and EGFR/KRAS/ALK-negative lung adenocarcinomas", Cancer Research (72: 1), pp. 100 –111.

[38] W. Pao, N. Girard (2011), –New driver mutations in non-small-cell lung cancer", The lancet oncology (12:2), pp. 175–180.

[39] RS Herbst, JV Heymach, SM Lippman (2008), –Lung cancer", The New England Journal of Medicine (359:13), pp. 1367–1380.

[40] F. Janku, DJ Stewar, R. Kurzrock (2010), –Targeted therapy in non-small-cell lung cancer–is it becoming a reality?", Nature Review Clinical Oncology (7:7), pp. 401–14.

[41] G. Bronte, S. Rizzo, L. La Paglia, V. Adamo, S. Siragusa, C. Ficorella C (2010), –Driver mutations and differential sensitivity to targeted therapies: a new approach to the treatment of lung adenocarcinoma", Cancer Treatment Review (36: Suppl 3), pp. S21–29.

[42] DE Gerber, JD Minna (2010), –ALK inhibition for non-small cell lung cancer: from discovery to therapy in record time", Cancer Cell, pp.18:548–51.

[43] http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE31210 , last accessed on 6/18/2013.

[44] Q Zheng, XJ Wang (2008), –GOEAST: a web-based software toolkit for Gene Ontology enrichment analysis", Nucleic Acids Research 36(Web Server issue), pp.W358 – 363.

APPENDICES

APPENDIX A

LIST OF GENES IN THE CLUSTER

Table A.1: List of genes contained in Cluster 2 for the normal tissue dataset

ZHX3	TNS1	RLIM	MPDZ	PBX1	NR2F2
ABCC9	CCDC50	TCF4	TCF4	NT5C2	FAM69A
DCN	PER1	ITPR1	GAPDH	SETBP1	LATS2
EFEMP1	HBB	FERMT2	PTGIR	GYPC	FBXL7
DIXDC1	SSBP2	SETBP1	PKIG	TMOD1	LOC100506948
DZIP1	PLSCR4	DST	ZEB2	MPDZ	TCF4
ALS2CR4	FLT1	DCN	DNAJB4	GUCY1A3	NFIA
VLDLR	CNRIP1	ID2	HGF	NCALD	DNAJB4
PBX1	MYH10	NFIA	PRELP	FSTL1	FHL1
NEGR1	ENAH	NEXN	AKAP12	LSAMP	CD34
KIF26A	GSTM3	PDE2A	KCTD16	NFIA	CREB5
AKAP12	NEXN	TACC1	ZEB2	AKT3	ITM2A
FBLN1	PTPRD	CCDC50	CNKSR2	HSPB2	FEZ1
FGF2	CDH13	ITGA8	ECM2	LEFTY2	SLIT3
PRELP	CAB39L	RERG	ERG	THBD	ITGA8

SEPT8	RECK	TACC1	PGR	MYH10	SYNE1
TPPP	PAK4	VWF	LTBP4	TMOD1	FBLN5
SASH1	RHOJ	FXYD1	PDK4	AGTR1	FHL1
GHR	HBA1///HBA2	LPHN3	LPHN3	HBB	SVEP1
LYVE1	GPC3	TNXB	FABP4	FHL1	TNXA///TNXB
HBB	MAMD2	MFAP4	TNXA	TNXB	FHL1
FABP4					

Table A.2: List of genes contained in Cluster 3 for the normal tissue dataset

AHNAK	NFYB	NR2F1	GABARAPL1	RNF125	RNF144B
BMPR2	ARHGAP24	PHACTR2	PTRF	PRKCH	PDLIM2
HEG1	SNRK	MACF1	PHACTR2	KIAA1324L	ATXN3
BMP5	HUWE1	KCTD15	IL11RA	VAPA	DNAJC18
LMCD1	ARAP3	ARHGAP24	PHACTR2	PTPRM	ICAM2
PLCE1	QKI	ANKS1A	CDKN1C	LMO2	CRIM1
CCBE1	KCTD15	SNX1	QKI	RNF207	CDKN1C
ARHGAP31	PTPRG	MYLIP	MEIS1	SHROOM4	RHOJ
ATXN3	GATA2	ENG	CD93	STARD13	KCNJ8
FRMD3	PIK3R1	FRMD3	PRKD1	FLT4	KIF17
ADAMTS8	PRDM5	KCNJ8	PTPRG	NOTCH4	PTRF
ZEB1	HEG1	RUNX1T1	LRCH2	CDKN1C	SVEP1

CFL2	COL4A3	TIE1	SH3BP5	PKIA	PRKCH
DAPK2	ESAM	SPTBN1	WASF3	DLC1	E2F2
TAL1	CD93	WWC2	KIAA1324L	PTPRM	AKAP2/// PALM2-AKAP2
ERG	SH2D3C	PECAM1	FGD5	SPOCK2	FZD4
CDKN1C	GPR133	CLEC1A	TNS1	ADRA1A	PECAM1
CELF2	WWC2	CLIC5	RADIL	EML1	NOSTRIN
DAPK2	PKNOX2	ADRB2	BDNF	HOXA5	ANGPT1
COL13A1	CAV2	FRMD3	LIFR	QKI	GPR146
BMP5	LDB2	CCBE1	CORO2B	CAV1	VAPA
SASH1	ACVRL1	TGFBR3	LIFR	NTNG1	BAI3
STX11	NEBL	PTPRB	SNCA	TGFBR3	SHROOM4
GATA2	SEMA6A	LPHN2	CD36	ERG	CNTN6
NECAB1	TEK	S1PR1	LIFR	ID4	PDZD2
CDH5	FOXF1	RNF182	CLIC5	LIFR	SYNPO2L
ADAMTS8	TSPAN7	SLIT2	TAL1	PTPRB	FGFR4
KL	BMPER	CLIC5	CCBE1	SDPR	EDNRB
RXFP1	SPTBN1	GDF10	SDPR	FREM3	USHBP1
CD36	MASP1	ADAMTS8	EDNRB	EDNRB	GPIHBP1
TCF21	FIGF	CLEC3B	NCKAP5	RSPO1	LRRN3

GRIA1	FGFR4	SGCG	AGER	AGER	

Table A.3: List of genes contained in Cluster 4 for the normal tissue dataset

PTPRM	UACA	GABARAPL1/// GABARAPL3	INPP5A	KLHDC1	DCHS1
EXT1	TIMP3	ARHGAP24	TNS1	AKAP2///PALM2-AKAP2	PLAGL1
LRRFIP1	TBX5	DST	CBR4	ITGA1	PECAM1
PDE3B	PLAGL1	APBB2	DST	SIK2	EFHA2
AKT3	NR2C1	RORA	CACNA1D	PCDHB6	FANCD2
C20orf46	CAV1	CBFA2T3	NPNT	SVEP1	P2RY14
DST	FIGN	LIN7A	NR2F1	GRIA1	LRRFIP1
RYR2	C13orf15	PIK3R1	SH3BP5	DOCK4	AKAP2 /// PALM2-AKAP2
ITGA1	PLAGL1	RHOJ	FIGN	UACA	CRIM1
ACACB	LIMCH1	ACACB	PTPRD	PREX2	FHL1
ACACB	AFF2	DOK6	TBX2	ANO2	RUNX1
DOK6	TENC1	SKA3	KCNK3	ANGPT1	GRK5
DACH1	LIMCH1	TBX2	SASH1	PKNOX2	C13orf15
GPX3	WWC2	C13orf15	RHOJ	RAPGEF4	FAT3

SHANK3	ARHGAP6	TBX5	PPP1R14A	PELO	MYLK
GPR82	PKNOX2	GPX3	NPR1	DACH1	CDH19
PTPRB	CDH19	COL6A6	ACTN2	TCF21	LRRTM4
KCNK3	KCNK3	AGTR1	FAT3	AOC3	

Table A.4: List of genes contained in Cluster 2 for the KRAS positive dataset

UACA	NFYB	ADCY4	NR2F1	GABARAPL1	RLIM
GABARAPL1 ///GABARAPL3	MPDZ	INPP5A	KLHDC1	PBX1	NR2F2
EXT1	CCDC50	RNF125	ARHGAP24	NT5C2	ATP6V0A2
PHACTR2	TNS1	PCM1	SNRK	SETBP1	PHACTR2
KGFLP2	KIAA1324L	ATXN3	DIXDC1	SSBP2	SETBP1
HUWE1	PLAGL1	TMOD1	TBX5	IL11RA	LOC100506948

VAPA	PECAM1	PLSCR4	DNAJC18	APBB2	ARHGAP24
MPDZ	PHACTR2	PLCE1	SIK2	EFHA2	ANKS1A
CDKN1C	ALS2CR4	CRIM1	FLT1	RORA	CACNA1D
PCDHB6	SNX1	NFIA	VLDLR	MYLIP	ID2
HGF	MEIS1	C20orf46	SYNPO	COPB1	ATXN3
CBFA2T3	GATA2	PBX1	MYH10	NFIA	DST
FIGN	LIN7A	NR2F1	GRIA1	LRRFIP1	RYR2
PIK3R1	SH3BP5	TNIP1	FRMD3	NEGR1	PIK3R1
ENAH	FRMD3	DOCK4	PLAGL1	ADAMTS8	PRDM5
CYCS	NOTCH4	FIGN	LSAMP	CD34	RUNX1T1
LRCH2	GSTM3	CRIM1	ACACB	KCTD16	NFIA
COL4A3	ACACB	RHOJ	PREX2	SPTBN1	WASF3
FHL1	TACC1	ITM2A	KIAA1324L	ACACB	CNKSR2
ERG	SH2D3C	DOK6	FGD5	ANO2	DOK6
FZD4	TENC1	ITGA8	KCNK3	ANGPT1	CLEC1A
GRK5	TNS1	LEFTY2	ADRA1A	DACH1	PECAM1
TBX2	CAB39L	RERG	SASH1	ERG	PKNOX2
ITGA8	PKNOX2	TACC1	HOXA5	PGR	MYH10
SYNE1	FRMD3	PAK4	GPR146	BMP5	LDB2
VWF	RHOJ	RAPGEF4	FAT3	SHANK3	VAPA

SASH1	TGFBR3	TMOD1	ARHGAP6	BAI3	SASH1
TBX5	RHOJ	NEBL	PPP1R14A	PTPRB	SNCA
RBPMS	PELO	TGFBR3	SHROOM4	GATA2	MYLK
FXYD1	SEMA6A	PDK4	LPHN2	ERG	NECAB1
PKNOX2	GPX3	AGTR1	NPR1	TEK	FHL1
DACH1	CDH19	S1PR1	GHR	PTPRB	ID4
PDZD2	CDH5	FOXF1	RNF182	ADAMTS8	LPHN3
LPHN3	CDH19	TSPAN7	ACTN2	TAL1	PTPRB
TCF21	KL	LRRTM4	TNXB	EDNRB	KCNK3
TNXA /// TNXB	MFAP4	TNXA /// TNXB	AGTR1	TNXA /// TNXB	SDPR
FREM3	FAT3	FHL1	USHBP1	ADAMTS8	EDNRB
FHL1	GPIHBP1	TCF21	CLEC3B	AOC3	NCKAP5
LRRN3					

Table A.5: List of genes contained in Cluster 3 for the KRAS positive dataset

PTPRM	AHNAK	ZHX3	TNS1	DCHS1	ABCC9
TCF4	TCF4	BMPR2	PTRF	PRKCH	DCN
PDLIM2	HEG1	LATS2	AKAP2///PALM2-AKAP2	MACF1	RHOJ

FHL1	FERMT2	PTGIR	GYPC	FBXL7	PKIG
LRRFIP1	KCTD15	DST	DZIP1	ITGA1	LMCD1
PLAGL1	ARAP3	DST	DST	ZEB2	PTPRM
QKI	TCF4	LMO2	CCBE1	DCN	DNAJB4
KCTD15	GUCY1A3	QKI	RNF207	ARHGAP31	CNRIP1
PTPRG	RHOJ	CAV1	NCALD	DNAJB4	P2RY14
ENG	CD93	PRELP	IDS	STARD13	FSTL1
KCNJ8	PRKD1	AKAP2/// PALM2-AKAP2	ITGA1	NEXN	RHOJ
KCNJ8	PTPRG	PTRF	ZEB1	HEG1	UACA
KIF26A	PDE2A	CFL2	CREB5	TIE1	PKIA
PTPRD	NEXN	CD93	ZEB2	AKT3	FBLN1
PTPRD	CCDC50	AKAP2/// PALM2-AKAP2	PECAM1	HSPB2	FEZ1
TBX2	FGF2	CDH13	ECM2	SLIT3	PRELP
RADIL	RHOJ	EML1	THBD	MAMDC2	ADRB2
RECK	BDNF	COL13A1	CAV2	QKI	CCBE1
CORO2B	LTBP4	CAV1	ACVRL1	NTNG1	STX11
SLIT2	SVEP1	BMPER	LYVE1	CCBE1	FABP4

FABP4	SGCG				

Table A.6: List of genes contained in Cluster 4 for the KRAS positive dataset

SRSF3	RNF144B	COPZ1	ARHGAP24	PER1	EFEMP1
BMP5	PDE3B	NR2C1	EFEMP1	SHROOM4	NPNT
SVEP1	C13orf15	FLT4	KIF17	CDKN1C	LIMCH1
SVEP1	IFT57	SH3BP5	PRKCH	DAPK2	ESAM
DLC1	TAL1	WWC2	PTPRM	AFF2	CDKN1C
ID4	GPR133	LIMCH1	CELF2	WWC2	CLIC5
NOSTRIN	DAPK2	C13orf15	ANGPT1	LIFR	TPPP
GPX3	WWC2	C13orf15	LIFR	FBLN5	CD36
CNTN6	LIFR	CLIC5	LIFR	SYNPO2L	HBA1 /// HBA2
COL6A6	HBB	FGFR4	CLIC5	GPC3	SDPR
KCNK3	HBB	RXFP1	SPTBN1	GDF10	CD36
MASP1	HBB	EDNRB	FIGF	RSPO1	GRIA1
FGFR4	AGER	AGER			

APPENDIX B

GO GRAPH FOR CLUSTERS

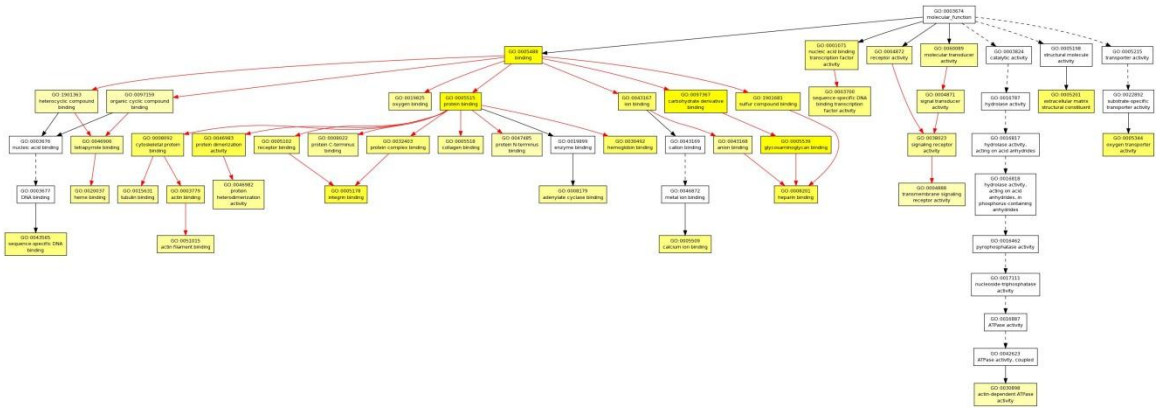

Figure B.1: GO graph for cluster 2 of normal tissue data set.

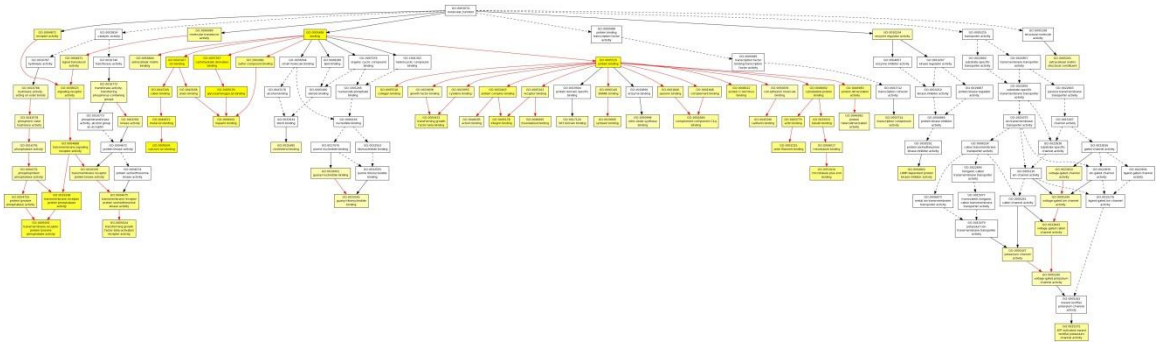

Figure B.2: GO graph for cluster 3 of KRAS positive data set.

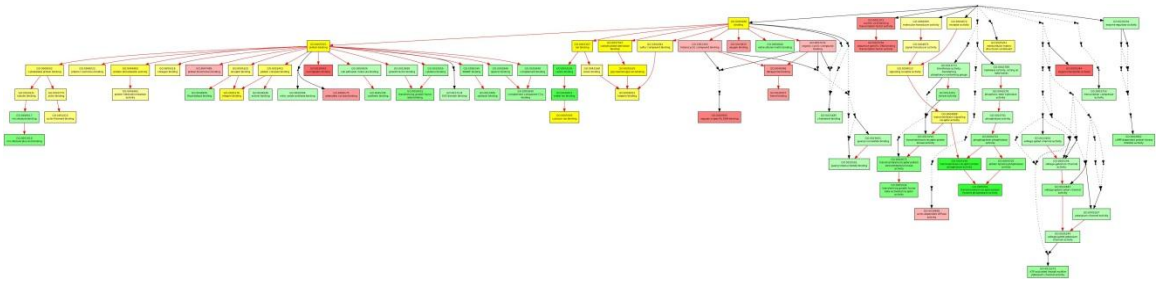

Figure B.3: Comparative GO graph for comparing GO enrichment status of Cluster 2 of normal tissue dataset and Cluster 3 of KRAS positive dataset.

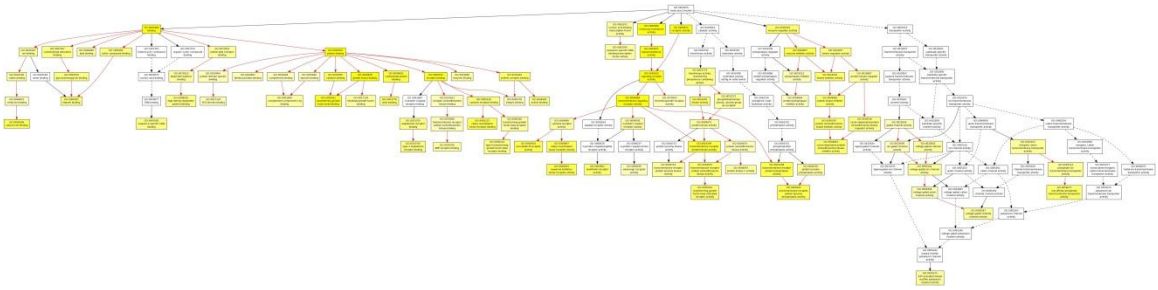

Figure B.4: GO graph for cluster 3 of normal tissue data set.

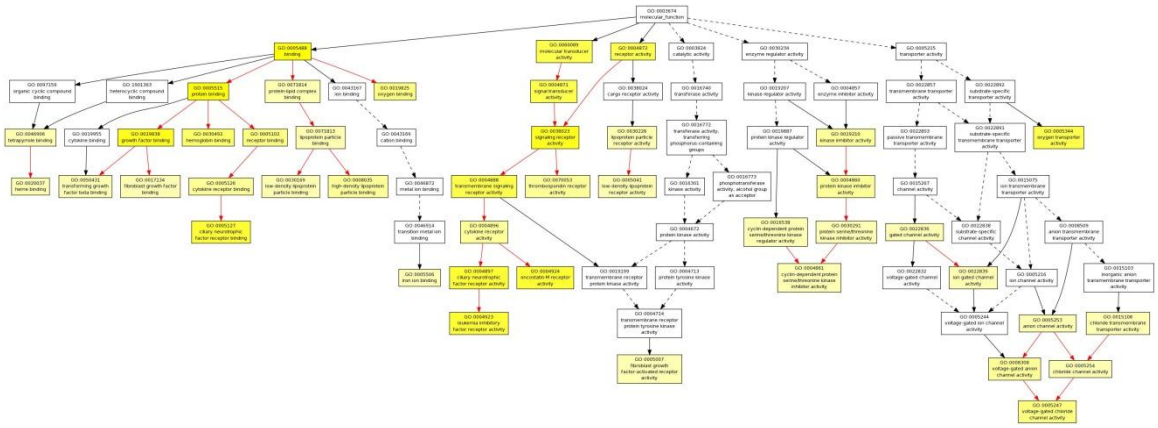

Figure B.5: GO graph for cluster 4 of KRAS positive data set.

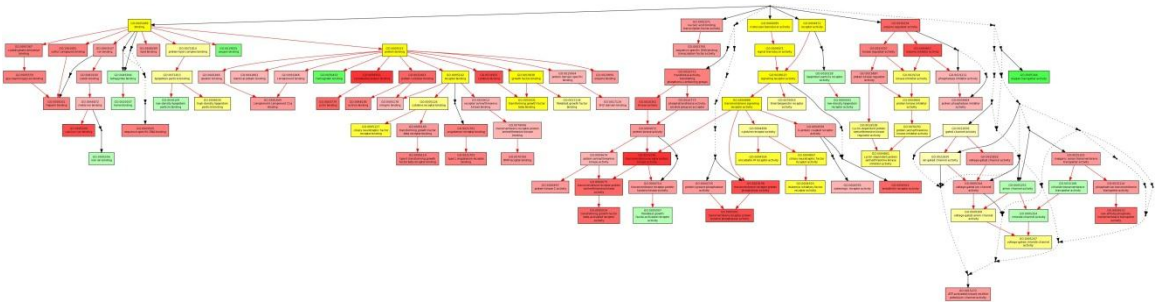

Figure B.6: Comparative GO graph for comparing GO enrichment status of Cluster 3 of normal tissue dataset and Cluster 4 of KRAS positive dataset.

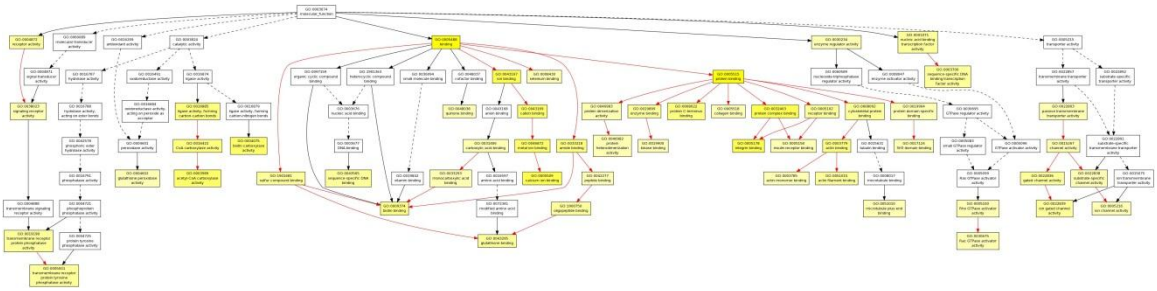

Figure B.7: GO graph for cluster 4 of normal tissue data set.

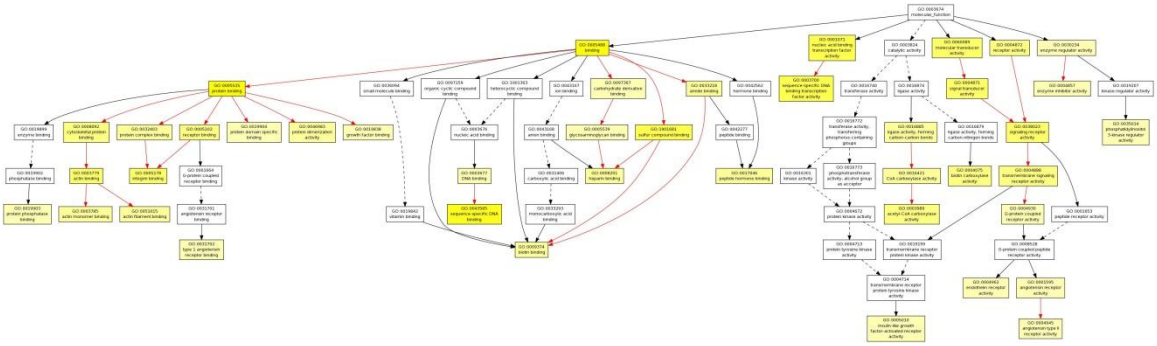

Figure B.8: GO graph for cluster 2 of KRAS positive data set.

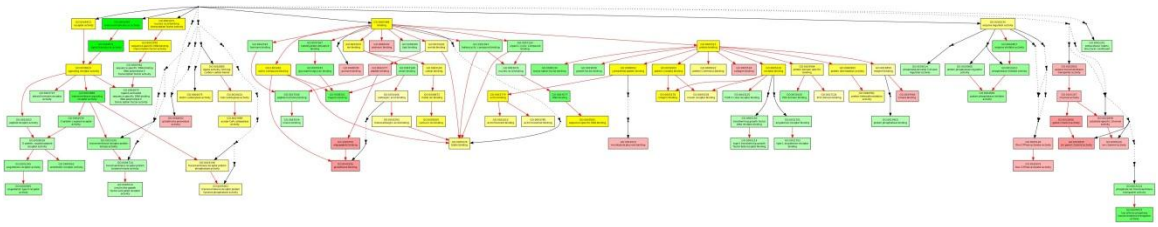

Figure B.9: Comparative GO graph for comparing GO enrichment status of Cluster 4 of normal tissue dataset and Cluster 2 of KRAS positive dataset.