1-1-2011

A Minimum Spanning Tree Based Clustering Algorithm for High throughput Biological Data

Harun Pirim

A MINIMUM SPANNING TREE BASED CLUSTERING ALGORITHM

FOR HIGH THROUGHPUT BIOLOGICAL DATA

By

Harun Pirim

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosopy
in Industrial Engineering
in the Department of Industrial and Systems Engineering

Mississippi State, Mississippi

April 2011

A MINIMUM SPANNING TREE BASED CLUSTERING ALGORITHM

FOR HIGH THROUGHPUT BIOLOGICAL DATA

By

Harun Pirim

Approved:

Burak Eksioglu Associate Professor of Industrial and Systems Engineering (Major Professor)	Mingzhou Jin Associate Professor of Industrial and Systems Engineering (Committee Member)
Allen Greenwood Professor of Industrial and Systems Engineering (Committee Member)	Andy D. Perkins Assistant Professor of Computer Science and Engineering (Committee Member)
Cetin Yuceer Assistant Professor of Forestry (Committee Member)	John M. Usher Professor of Industrial and Systems Engineering (Graduate Coordinator)
Sarah A. Rajala Dean of the James Worth Bagley College of Engineering	

Name: Harun Pirim

Date of Degree: April 29, 2011

Institution: Mississippi State University

Major Field: Industrial Engineering

Major Professor: Dr. Burak Eksioglu

Title of Study: A MINIMUM SPANNING TREE BASED CLUSTERING ALGORITHM FOR HIGH THROUGHPUT BIOLOGICAL DATA

Pages in Study: 96

Candidate for Degree of Doctor of Philosopy

A new minimum spanning tree (MST) based heuristic for clustering biological data is proposed. The heuristic uses MSTs to generate initial solutions and applies a local search to improve the solutions. Local search transfers the nodes to the clusters with which they have the most connections, if this transfer improves the objective function value. A new objective function is defined and used in the heuristic. The objective function considers both tightness and separation of the clusters. Tightness is obtained by minimizing the maximum diameter among all clusters. Separation is obtained by minimizing the maximum number of connections of a gene with other clusters. The objective function value calculation is realized on a binary graph generated using the threshold value and keeping the minimum percentage of edges while the binary graph is connected. Shortest paths between nodes are used as distance values between gene pairs. The efficiency and the effectiveness of the proposed method are tested using fourteen different data sets externally and biologically. The method finds clusters which are similar to actual ones using 12 data sets for

which actual clusters are known. The method also finds biologically meaningful clusters using 2 data sets for which real clusters are not known. A mixed integer programming model for clustering biological data is also proposed for future studies.

Key words: clustering, optimization, heuristics, networks, integer programming

DEDICATION

To Yunus, the denouement of the Starkville story...

ACKNOWLEDGMENTS

After all, I thank to having a little bit feeling of Isaac Newton's quote: "I do not know what I may appear to the world, but to myself I seem to have been only like a boy playing on the sea-shore, and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay all undiscovered before me".

I thank my wife Suendam Birinci Pirim and our parents for their spiritual and physical support. I thank my committee members for their valuable comments on this dissertation, and I especially thank Dr. Burak Eksioglu for directing this research and the tolerance he is gifted.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Clustering in biology has a history that goes back to Aristotle's attempt to classify living organisms [6]. Today, clustering genomic data stands out as an approach to deal with high dimensional data produced by high throughput technologies such as *gene* expression *microarray*s [94]. Biological data were limited to DNA sequence data before the *genome* age in the 1980s [75]. Nowadays, terabytes of high throughput biological information are generated with the advent of new technologies, such as microarrays, *eQTL* mapping, and *next generation sequencing*. Now, a need for exploiting computational methods exists to analyze and process such amounts of data in depth and in different ways to address complex biological questions regarding gene functions, gene co-expression, protein-protein interactions (PPI), personalized drug design, systems level functional analysis of plants and animals, and organism-environment interaction. This fact has given birth to disciplines like bioinformatics, computational biology, and *systems biology*.

In physics, before mathematical models were incorporated; i.e., before Newton, the discipline was stamp collecting (i.e. descriptive). Incorporation of mathematical models changed physics into a predictive science. In a similar manner, incorporation of computation into biology is changing the discipline from being a descriptive science to a predictive science. One of the prediction methods used in biology to analyze the high throughput

data is clustering. As a data mining method, clustering of genomic data was well studied during the last decade. Clustering is also a well known and studied problem in the operations research (OR) field. However, clustering of genomic data is relatively not well studied by the OR community, although data mining techniques have been used in market segmentation and facility location problems, for example.

Moreover, aspects of biological theories can be modeled with OR tools. One of these aspects is that a small subset of genes are typically involved in a particular cellular process of interest, and a cellular process happens only in a subset of *sample*s [72]. Another aspect is that genes of the same pathway may be induced or suppressed simultaneously or sequentially upon receiving stimuli [163]. A third aspect is that most biologists assume an approximately *scale-free topology*, or a *small world property*, for networks constructed from *gene expression* data [159]. Hence, one may say that genes with high *connectivity* are much fewer in number than genes with low connectivity [144]. Thus, this chapter discusses many diverse approaches and algorithms that currently exist for clustering of genomic data from an OR perspective by introducing background in molecular biology, and presenting clustering approaches and techniques. The chapter is organized as follows: Section 2 gives concise information about molecular biology and relevant disciplines; Section 3 discusses the clustering of genomic data problem, and provides a problem definition and data representations; Section 4 reviews recent algorithms used for clustering genomic data; Section 5 concludes and suggests future research directions for the operations research community; and section 6 provides the glossary that includes definitions of the italicized words and phrases throughout the text.

1.1 Background Information about Molecular Biology

The essential cellular molecules for a biological system to function and interact with its surrounding include DNA, RNA, proteins, and *metabolite*s, all of which are under physiological and environmental control. Many different interaction layers exist among these molecules such as PPI networks, i.e., interactomes, gene regulatory networks (GRNs), biochemical networks, and gene co-expression networks. A holistic picture of these interactions is being studied through systems biology.

Based on the central dogma of molecular biology, DNA transcribes into RNA, and RNA translates into proteins, some of which then serve as catalysts in the production of metabolites. A gene is expressed upon receiving the transcriptional signal. Genes have *activator*s and *repressor*s. Genes reveal no or low expression values without activators. Repressors block gene expression, even in the presence of activators. Transcription factors (TFs) are activator or repressor proteins produced by genes. TFs bind to *regulatory site*s and turn them on to transcribe RNA or off. Genes may show cascade interactions. For example, the product of one gene may increase or decrease the transcription rate of the other, and this process may continue downstream including temporal or causal order of molecular events.

It is often preferred to analyze thousands of genes' dynamics together rather than one at a time. The DNA microarray (Figure 1.1) has been one of the commonly used technology to measure thousands of gene expressions simultaneously[94], and microarray data have been stored in public databases such as the Gene Expression Omnibus (GEO) for further analysis. For example, Affymetrix GeneChip Mouse Genome 4302.0 Array provide

45,000 probe sets to analyze expression levels of more than 39,000 transcripts. *Feature* size is 11 μM. 11 probe pairs per sequence are used.

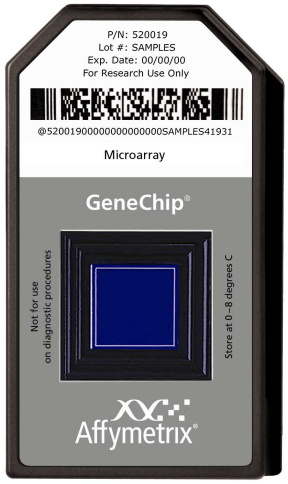

Figure 1.1

A microarray chip produced by Affimetrix, courtesy of Affymetrix.com

The data extracted from microarrays or a similar technology is analyzed using a *reverse engineering* approach. A simplified framework of reverse engineering methodology for modeling GRNs from gene expression data is shown in Figure 1.2, which is adapted from [85]. However, it is a challenging task to infer about GRNs because expression data are high-dimensional, complex, and non-linear. Further complicating the inference is that, dynamic relations exist among thousands of genes, expression data involve *noise*, and the sample-gene ratio is normally very small [161] since the array chips corresponding to samples are expensive. Co-expressed genes show coherent *expression pattern*s, indicating that they may have similar functions [94] or co-exist in a pathway. However, different

external conditions may trigger a gene to be expressed similarly with different group of genes [94]. Genes with similar expression patterns are more likely to regulate each other or to be regulated by a parent gene [104]. Here, the problem of quantifying the relations between genes arise.

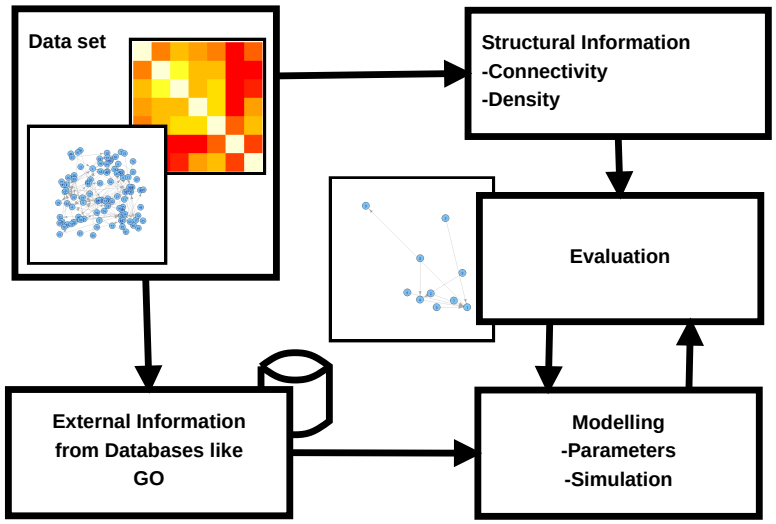

Figure 1.2

Reverse engineering to infer about the extracted data

A powerful clustering approach as well as a predictive model may detect patterns or relationships in expression data [94]. However, a predictive model should be guided by biological facts, meaning that results of predictive models should be validated by biological knowledge. On the other hand, biological experiments should be guided by computational methods to make the best use of biological data and reduce experimental and time costs (Figure 1.3). Online databases exist to facilitate *validation* of the results obtained from

predictive models. Incorporation of the database knowledge to modeling GRNs is essential for more accurate results or for comparing the model to reality.

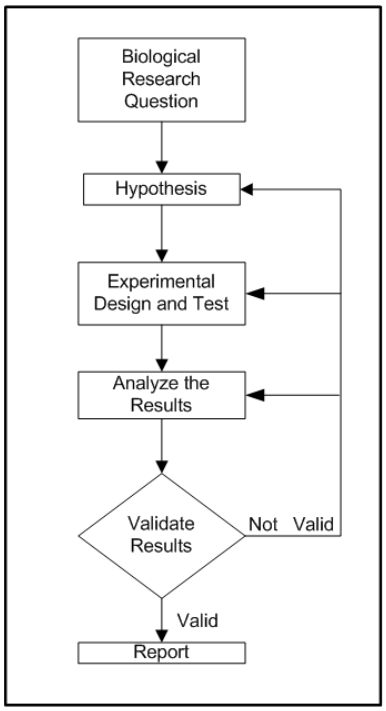

Figure 1.3

Biological experiment and validation work flow

1.2 Problem Definition and Representations of Genomic Data

Clustering generates individual groups of data called a *partition*, rather than assigning *objects* into the groups already known as in *classification* [9]. A partition is defined as follows:

$P = \{c_1, c_2, ..., c_s\}$ where s is the number of clusters.

$\sum_{i=1}^{s} |c_i| = n$ where n is the number of objects and $|c_i|$ is the cardinality of *cluster* i

$X = \{x_1, x_2, ..., x_n\}$ is the set of n objects and $Y = \{y_1, y_2, ..., y_n\}$ is the set of n patterns where $y_i \in R^d$ and d is the number of samples. The clustering problem is finding a partition that has clusters with objects having similar patterns.

There is no universally accepted definition of a cluster. However, objects in a cluster should be similar or coherent and objects in different clusters should be dissimilar. In other words, similarity within a cluster is maximized, and similarity between clusters is minimized.

Clustering is often used in the genomic data analysis process. Genomic data analysis is an integrated process that comprises low-level and high-level analysis. Cluster analysis for genomic data consists of three main steps: 1) pre-processing the data so that the clustering algorithm can use the data as an input; 2) using a clustering algorithm with an appropriate distance measure; and 3) using an index and/or *biological database* to validate the quality of the clusters found. *Data pre-processing* is essential before clustering, since it affects clustering results. The effects of normalization and pre-clustering techniques have been demonstrated on clustering algorithms [133], so have the effects of filtering methods [140]. The distance measure can also affect the results from a clustering algorithm [62].

Although there are many problems associated with cluster analysis and there are many biological data types, this chapter mainly focuses on clustering algorithms as applied to microarray data unless otherwise mentioned. As an illustrative example, we use a breast cancer microarray data set. The data set is pre-processed [157]. Then 49 samples corresponding to 4 different collection of tumors consisting of 1213 genes each is used. The pre-processed expression image is shown in Figure 1.4. Color densities and corresponding

expression values are shown on the right vertical color bar of the Figure. The samples are shown on the y axis while the genes are shown on the x axis.

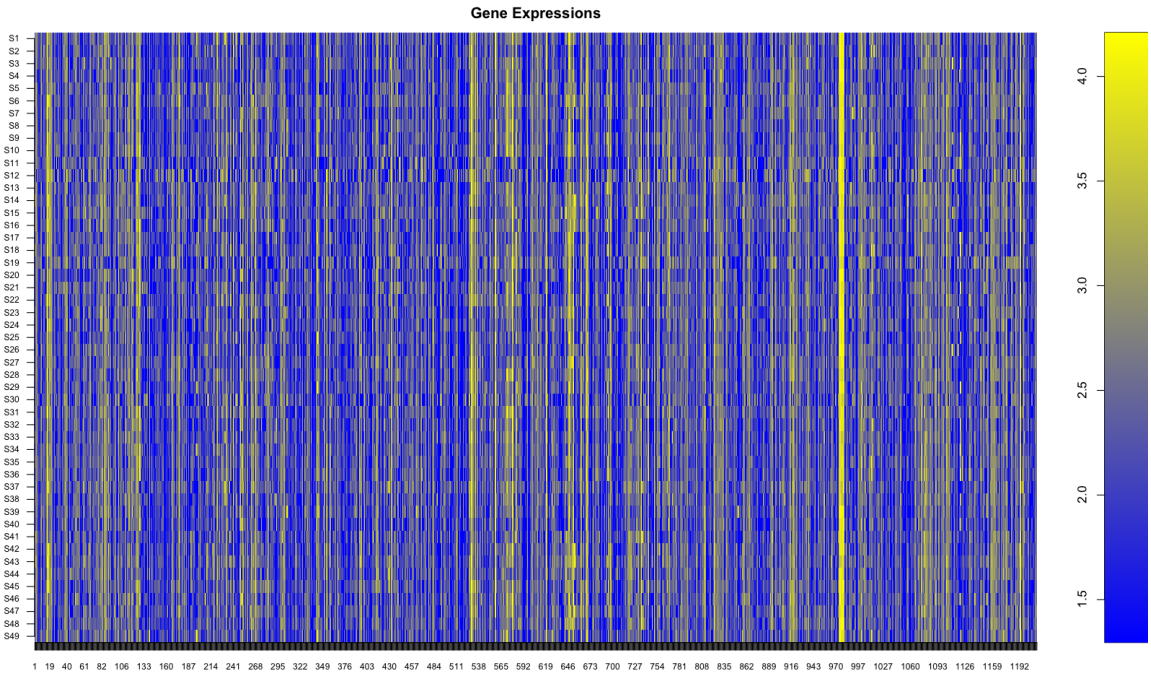

Figure 1.4

Image plot of expression values

Since the real partition of the samples is known, clustering of samples is desired for the purpose of external validation. K-means (see section 4.1) as applied in R base package is chosen for clustering. The *Eucledian distance* matrix between samples and the number of clusters, i.e., 4, are inputs to the K-means algorithm. The partition generated by K-means and the real partition are shown in Table 1.1. It should be noted that the order of the numbers identifying clusters of the real partition may not be the same in the generated partition. The last step of the cluster analysis is validation using the *C-rand* index. The C-

rand value found is 0.343. This means that K-means could not find a partition very similar to the real one since the best C-rand value would be 1.

Table 1.1

Partition of samples, S1:S49, into four clusters

1 to 24	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S23	S24	
K-means	4	4	4	2	1	1	2	2	2	2	3	3	1	1	1	1	4	4	3	4	4	2	1	1	
Real	1	1	1	4	4	4	4	1	4	4	3	3	4	4	4	4	3	3	3	1	3	4	4	4	
25 to 49	S25	S26	S27	S28	S29	S30	S31	S32	S33	S34	S35	S36	S37	S38	S39	S40	S41	S42	S43	S44	S45	S46	S47	S48	S49
K-means	4	1	1	1	2	3	2	4	4	2	2	2	1	4	4	4	1	1	1	1	1	2	1	1	2
Real	1	4	4	4	4	3	2	1	1	1	2	2	2	1	1	1	2	2	2	4	2	4	2	2	2

1.2.1 Quantification of Relations

Distance measures are used for defining relationships between the biological molecules of interest. Clustering algorithms use this relationship in different ways. Hoeffding's D meausure outperforms the others in quantifying non-linear associations when Person correlation, Spearman correlation, and Hoeffding's measure were compared for gene expression association analysis [45]. Bandyopadhyay and Pal [13] propose new distance measures based on Euclidean and *Manhattan distance* measures where *normalization* is dependent on the experiment type, i.e., samples. Balasubramaniyan et al. [10] also use a local shape based distance metric based on Spearman rank correlation. The metric is used to identify local similar regions in gene expression profiles.

Pairwise relations between genes are often preferred for quantification, because it is computationally less costly than stochastic approaches where a relation is considered con-

ditionally to other relations. Correlations or distance measures, e.g., Euclidean distances between gene pairs are calculated using the expression data, and then the resulting data matrix is used in a clustering algorithm to find the clusters of genes. However, use of direct distance measures between pairs of genes is somewhat traditional as opposed to transitive distance measures used between genes. Traditional use of a distance measure employs the "Guilt-by-Association" assumption that genes having similar expression values generally have similar functions and the genes with dissimilar expression values do not have similar functions [164]. The traditional approach is "Guilt-by-Association" because a biological function is often the result of many genes interacting with each other rather than a result of a simple pairwise relation [164]. However, transitive distance implies that there is at least one path, not necessarily of length 1 as in a pairwise relation, between two genes, and the length of this path is the distance between them. Researchers proposed that a transitive co-expression analysis applying a shortest path distance between two genes (Figure 1.5) gives biologically meaningful results, rather than a direct pairwise distance measure [162, 164]. Zhu et al. [164] use a hybrid distance matrix having both direct and shortest-path distances for *clustering*. Phan et al. [116] also use transitive directed acyclic graphs for representation of expression patterns. Once the data are clustered based on a distance measure, validation of the clustering algorithm's performance is essential.

1.2.2 Validation of the Partitions

Before dealing with validation of the partitions generated by clustering algorithms, there are sub-problems to consider: *filtering* mechanisms to be used for the data, algo-

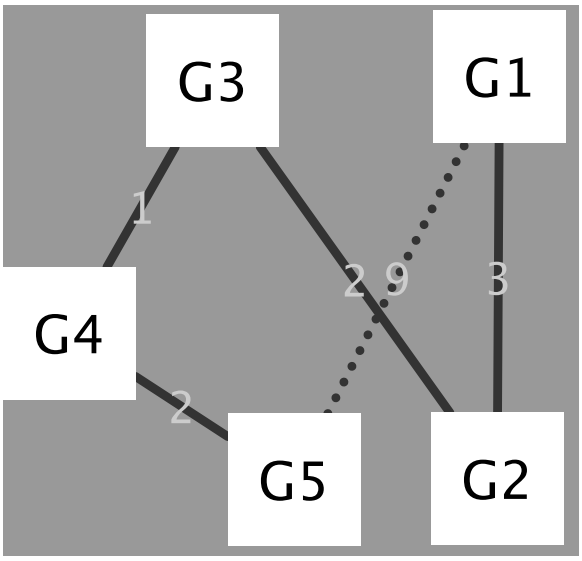

Figure 1.5

Transitive distance - the distance between genes G1 and G5 is 8 rather than 9

rithm to be used, the number of clusters, distance metric to be used if it is used by the clustering algorithm, cut-off height (level) for the *dendrogram* of genes in case a hierarchical clustering is used, approach to be used like aggloramative or divisive, validation methods, and measures for generated clusters. These are some of the aspects that affect validation results.

Outputs of clustering algorithms need validation to check whether the genes in the same clusters have biological relations or not. Clusters should make sense biologically. Clusters should be reliable, not formed by chance. The stability of a clustering algorithm, the validation of the generated cluster using *biological database*s, and the comparison with other algorithms are important aspects to measure reliability. Stability can be assessed by

both sensitivity of the algorithm to the user-specified parameters and small modifications to the data sets [4].

There are mainly four different ways to validate the performance of a clustering algorithm: 1-Visual validation: inspects if the algorithm detects a special structure of the data, e.g., number of clusters may be detected on the 2D graphics. For example, Figure 1.6 implies that the optimal number of clusters is two; 2-External validation: requires the knowledge of the real partition, e.g., C-rand or pre-defined structure of the data. 3-Internal validation: uses the features of the partition such as compactness, e.g., ensuring that variance within clusters are small and examining the separation of clusters, e.g., single linkage, average linkage, complete linkage; 4-Biological validation: uses biological annotations to see if the genes in clusters are enriched for biological terms significantly.

Each clustering validation technique has its own bias towards a given clustering criterion [41]. Ensemble and multi objective clustering approaches [41] are used to address the problem of being biased towards a particular objective or a clustering criterion. A good clustering algorithm may or may not depend on prior knowledge, or many user-defined parameters. Jiang et al. [72] propose that the algorithm should be able to extract useful information, detect the embedded and highly connected structure of genomic data, and provide graphical representation of the cluster structure. Functions of some genes are published in relevant databases and genes with known similar functions may guide the clustering by being assigned to the same cluster. This partial knowledge can also be used as an input for a clustering algorithm with the expectation that the resulting clusters will be more biologically meaningful [72]. For example, Cohen et al. [31] propose an algorithm

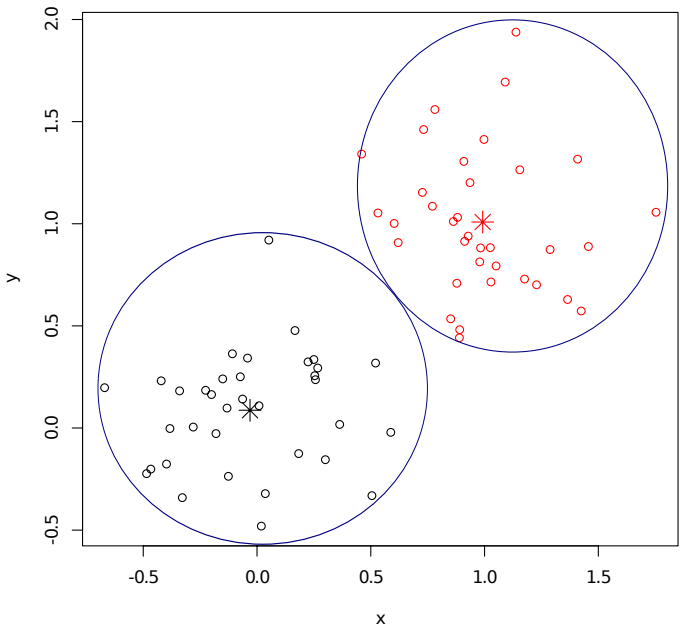

Figure 1.6

2D Graphics of Clusters Generated by K-means

that integrates semantic similarities from ontology structure to the procedure of getting clusters out of a dendrogram.

1.2.3 Representation of Expression Data

Gene expression data is usually represented as an $n \times m$ matrix where n is the number of genes and m is the number of time points or samples. Microarray features, or gene transcripts, are the rows of the expression matrix and are represented as vectors. Gene expression data sets are comprised of gene expression levels over time points, also called time course data (Table 1.2), or samples, such as control vs. treated. Clustering may be performed by grouping genes over samples or samples over genes. Since the number of genes is normally thousands and many of the genes have low or invariant expression

values, filtering gene expression data to reduce the dimension of the $n \times m$ matrix is often necessary. Gene interactions may be represented by graphs using an adjacency matrix. A graph G consists of vertices $V(G)$ that represent genes, edges $E(G)$ that represent relations between genes. Assuming a loopless, simple graph adjacency matrix $A(G)$ has elements $a_{i,j}$ equal to 1 if i has relation with j, 0 otherwise. If the corresponding graph is not relational, i.e, binary then a weight $w_{i,j}$ is associated with the edges showing the strength of the relation between i and j.

Table 1.2

A Sample Microarray Data [69]

Gene Name	0HR	15MIN	30MIN	1HR	2HR	4HR	6HR
EST W95908	1	0.72	0.1	0.57	1.08	0.66	0.39
SID487537 EST AA045003	1	1.58	1.05	1.15	1.22	0.54	0.73
SID486735	1	1.1	0.97	1	0.9	0.67	0.81
Genes	.	.	.	.	.	.	.
	.	.	Expression Values			.	.
	.	.	.	.	.	.	.
MAP kinase phosphatase-1	1	2.09	3.37	5.52	4.89	3.05	3.27
MAP kinase phosphatase-1	1	1.52	4.39	7.03	5.45	2.93	3.91
MAP kinase phosphatase-1	1	2.25	4.67	7.94	5.94	3.76	4.46

Clusters are generated by clustering algorithms that use a data representation as an input. The way the biological data is represented, whether it be a network, matrix, vector, may ease the computation for the problem on hand. For instance a naive hierarchical

clustering (HC) algorithm has time complexity of $O(n^3)$, however the time complexity may be reduced to $O(n^2 \log\ n)$ using a *priority queue* data structure [96]. Representation of gene expression data as an $n \times m$ matrix or network may help a researcher focus on the genes of interest by making use of matrix theory and graph theory.

Complex interactions between molecular components of a biological cell are sometimes modeled with graph structures to get support from graph theory. Visualization and computational representation of these interactions as networks enables wide range of applications [131]. Models of GRNs fall between abstractness like Boolean networks, or relevance networks, and concreteness, including biochemical interactions with stochastic kinetics [85]. Abstract models are scalable to large networks but are further from reality whereas concrete models are not scalable to large networks but more accurately reflect biological reality. Hence there is a trade-off between scalability and concreteness. Network models can be discrete or continuous. Deterministic or probabilistic Boolean networks and Bayesian networks have discrete variables whereas the neural network models and differential equations based models use continuous variables. Abstract networks such as co-expression networks use edges from hypothetical inference, whereas concrete ones such as PPI use edges inferred from physical interactions [164]. Chen et al. [24] construct a network for experimentally detected PPI. Nodes represent proteins and edges are the interactions with edge weights calculated based on a predefined formula. The authors propose a novel measurement to assess the reliability of PPIs using topological features of the network, since PPI data involves high false positive rates and also develop an algorithm to measure reliability efficiently in PPI networks.

There may be different relations between the molecular components. For instance, the components may interact with each other, one of the components may regulate the expression of the other, inhibit, or stimulate the activity of the other [38]. All these relationships can be represented using networks, or graphs. Graph structures are used to suggest some biological questions about discovering potential drug targets. Graph topology reflects functional relationships and neighborhoods of genes [38]. Network models are a very popular way of formalizing available knowledge of cellular systems in a consistent framework [16]. For instance, *factor graph*s are minimal graphs for inferring expression data [16]. Expression data may be integrated with *transcription factor* (TF) binding data to further infer interaction networks, and time course expression data may be integrated with physical interaction networks to identify pathways [16].

1.3 Algorithms Used for Clustering Genomic Data

The algorithms used in clustering gene expression data are usually grouped into two classes: partitional and hierarchical. However, clustering algorithms may also be grouped based on the representation of data, relationship between clusters, distribution of the data, and other properties. For example, some of the classes of algorithms include flat, or partition based clustering, hierarchical clustering, biclustering, model based clustering, metaheuristic clustering, fuzzy clustering, optimization based clustering, network based clustering, and ensemble clustering. Of course, these groups may have intersections, and there may be hybrid approaches Chipman and Tibshirani [26]. Clusters may be exhaustive, meaning that each object is assigned to a cluster, or non-exhaustive, meaning that

some objects may be assigned to no cluster. Exclusive clusters are non-exhaustive ones to which an object is either assigned or not [96]. Objects are assigned solely to one cluster in hard clustering; whereas soft clusters, sometimes called overlapping clusters, may have common objects with non-negative value memberships. For different definitions of hard, soft, and partitional clustering see [96]. Different types of clustering algorithms are defined based on diverse features, such as representation of data, relation between clusters. The following subsections reviews the most recent and common methods.

EBSCO host and PubMed databases were investigated for obtaining the articles used in the review. However the articles utilized were not limited to these databases. "Clustering method" and "microarray data" or "gene expression data" inputs were used in EBSCO host. There were 250 results, 29 relevant. "Clustering of gene expression data" input was used in PubMed. 6706 results were pulled. The results were filtered based on being recent, i.e., after 2005, and having potential contribution to the review being comprehensive enough. More than 100 articles were used for the review. The following sections present classifications and review clustering algorithms used for biological data analysis based on the papers from the databases. Since one of our objectives is to increase the interest of OR researchers, more details are provided on some classes of algorithms, such as optimization based one.

1.3.1 Flat Clustering Algorithms

In flat clustering, objects are partitioned based on a (dis)similarity metric. K-means is perhaps the most widely used method. K-means is a randomized algorithm which gener-

ates cluster centers randomly and assigns objects to the nearest cluster center. The algorithm modifies the location of the centers to minimize the summation of squared distances between objects and their closest cluster centers. Richards et al. [119] report that K-means performed faster and resulted in more biologically enriched clusters compared to three other methods. On that study K-means was used to cluster human brain expression data sets which had approximately 20,000 genes and 120 samples. Bohland et al. [17] use K-means to cluster all left hemisphere brain voxels, $25,155 \times 271$ matrix is used as an input for the algorithm. Sharma et al. [129] use a two-stage hyperplane algorithm applied in a software package called HPCluster. The first stage reduces the data and the second stage is the conventional K-means. The algorithm can handle 44,460 genes without failure. [142] develop a clustering method which doesn't force all the genes into clusters. The method employs a truncation of the clustering tree first, and then applies the K-means algorithm to avoid K-means being trapped in local minimum. The method is applied on both simulated and embryonic stem cell data. The authors supply a C library and a package to implement the method and visualize data. Tseng [141] develops a K-means derivative, applying a penalty to avoid scattered objects being assigned into clusters and weights to incorporate prior information. The developed method is used for both mass spectrometry and microarray data sets.

K-means requires specification of the number of clusters before clusters are generated. K-means is also sensitive to noise that is prevalent in gene expression data [72]. Furthermore, a partition generated by K-means may not be globally optimum since it relies on randomly chosen initial objects. Hence K-means is sensitive to initial partitions; it may

be trapped in local optima; and it is applicable to data with only spherical-shape clusters [149], which is not always the case for biological data. The time complexity of K-means algorithm is $O(i\ k\ n\ m)$ [96] where i is the number of iterations, k is the number of clusters, n is the number of objects and m is the dimension of an object.

Partitioning Around Medoids (PAM) [76] is also a widely used flat clustering algorithm. PAM computes medoids for each cluster. PAM is computationaly more costly than K-means since it requires pairwise distance calculation in each cluster. Wang et al. [145] use the system evolution principle of thermodynamics based on PAM to predict the number of clusters accurately. Huang and Pan [66] incorporate a gene's function knowledge into a new distance metric. Distances between genes with known similar function are shrunk to 0 before the genes are clustered using K-medoids or the PAM algorithm; then, remaining genes are assigned to existing clusters and/or new clusters.

Self-Organizing Map (SOM) is another flat clustering approach based on neural network methods widely used in gene clustering. Ghouila et al. [48] employ a multi level SOM based clustering algorithm in the analysis of macrophage gene expression data. SOM also requires the number of clusters and the grid structure of neurons as inputs. SOM maps high dimensional data into 2D or 3D space. The potential of merging distinct patterns into a cluster can make SOM ineffective [72].

Knowing or predicting the number of clusters correctly for a flat clustering algorithm affects the quality of the clusters. Jonnalagadda and Srinivasan [73] develop a method to find the number of clusters in gene expression data. They evaluate different partitions from

a clustering algorithm and find the partition that describes the data best. They use an index measuring information transfer for additional clusters.

1.3.2 Hierarchical Clustering Algorithms

Hierarchical clustering (HC) algorithms generate dendrograms that show relationships of objects and clusters as hierarchies (Figure 1.7). HC algorithms can be divided into two groups: agglomerative and divisive. In agglomerative clustering, all the objects begin in individual clusters. Then, the object pair with the highest similarity is found and merged to be included in the same cluster. The objects then merge, or agglomerate iteratively, until only one cluster exists which includes all the objects. The merging process can be stopped at any time with a stopping criterion. A complete run of an agglomerative clustering algorithm produces a complete graph where each node has relations with other nodes and a dendrogram where relationships between objects appear. Divisive HC methods work contrary to agglomerative HC methods. Divisive clustering methods iteratively divide the complete graph into smaller components by finding the pair of objects that have the lowest similarity and removing the edges between them. Divisive clustering can be represented by a dendrogram that gives smaller components at each successive split of the network. The dendrogram's branches are the clusters. These branches also give information about similarity between clusters.

Level Selection Methods One challenge encountered in HC is selection of the level that is used to cut the dendrogram through a number of branches corresponding to the number of clusters. Wild and Blankley [148] test nine cluster level selection methods based on

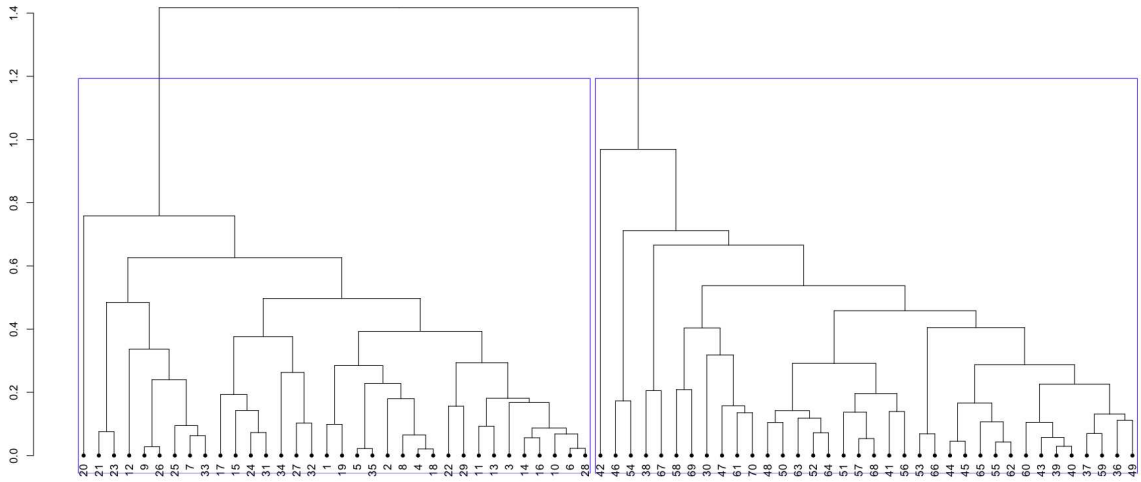

Figure 1.7

Dendrogram of the data generated for Figure 1.6

their lack of parametrization and simplicity. Neither of these methods outperform the others consistently on all data sets used. Kelley et al. [77] present an automated method for cut-off level selection to avoid the dangers of using a fixed valued cut-off. Zahoránszky et al. [158] present a new cluster selection method for HC. The method does not require a similarity measure and is suitable for data with a graph representation. It relies on cohesive clusters in which all pairs of objects are similar to each other.

Langfelder et al. [82] propose an algorithm that defines clusters from a hierarchical tree. However, they overcome the inflexibility of the fixed-height cut-off choice of the dendrogram. Their algorithm adapts to the shape of the dendrogram, is capable of detecting nested clusters, and can combine the advantages of hierarchical clustering and PAM. However, it is stated that optimal cutting parameters and estimation of number of clusters in the data set are still open research questions. They apply the algorithm on both hu-

man gene expression and simulated data. Although the algorithm has many user defined parameters, it is reported that it works well with default settings compared to PAM and normal HC.

There are a number of HC applications for biological data. Liang and Wang [88] propose a dynamic agglomerative clustering method and apply this on leukemia and avian pineal gland gene expression data. The numerical results show that the proposed method is convenient for data sets with or without noise, which is defined as scattered, singleton or mini-cluster genes. The method collects scattered genes in a cluster and groups other clusters dynamically and agglomeratively.

HC algorithms are not robust to noise, and they have high computational complexity [72] which is $O(n^3)$ [96] where n is the number of objects. They are "greedy," meaning they combine the most similar two objects at the first step, and the following steps are affected by the initial step and so on.

HC and K-means algorithms introduced in the previous section are root algorithms upon which many algorithms are built. Comparison guides the choice of the clustering algorithm [139, 137]: one should look at root clustering approaches and the desired features required for the application in which one of the root approaches is used. A review of root clustering approaches, partitional, K-means, or hierarchical and improved algorithms based on the root approaches are presented in [6, 36, 33].

1.3.3 Network Based Clustering Algorithms

HC algorithms make use of data represented as networks. However, network based clustering algorithms are not all hierarchical. As mentioned earlier, biological data may be represented using networks. Hence, many clustering algorithms use network data structures to cluster biological data sets. For example, gene expression data may be regarded as a complete network where the genes are the nodes of the network, and pairwise correlation values obtained from expression data are the edge weights of the node pairs. Hence, clustering this network data is a graph partitioning problem. Algebraic graph theory may be employed for the purpose of clustering a network. One algebraic graph theory tool is spectral clustering, a form of graph partitioning where the eigenvalues and eigenvectors in the Laplacian matrix, the difference between the adjacency and degree matrices, are usually used to reduce the dimension of the similarity matrix. The new matrix with reduced dimensions is used as an input for K-means or another algorithm [79]. Higham et al. [60] formulate a discrete optimization problem that results in a class of spectral clustering algorithms. They test the performance of the spectral algorithms on three different microarray data sets involving different types of diseases. Higham and Kalna [59] present spectral analysis of *two-signed microarray expression data*. The time complexity of a general spectral clustering algorithm is $O(n^3)$ because of the eigenvalue computations.

Clustering based on each node's neighbors is also widely used for genomic data. Huttenhower et al. [68] propose a graph based clustering algorithm called nearest neighbor networks (NNN). This algorithm first generates a directed graph with each gene connected to a specified number of nearest genes. Then, the graph is converted to an undirected one

by keeping only the genes having a bidirectional relationship. Overlapping cliques of a specified size are merged to produce preliminary networks. Then, the preliminary networks containing cut-vertices are split, keeping the copies of the cut-vertices. They also introduce a software implementation of the algorithm proposed. Mete et al. [103] propose an algorithm to find functional modules from large biological networks. The algorithm assigns nodes to the same cluster based on how they share common neighbors. Using three steps, the algorithm detects clusters, *hub*s, or most connected nodes, and outliers of the network. The first step checks every vertex for being core, having a defined number of neighbors, or not. If it is a core vertex, a new cluster is expanded. Otherwise, the vertex is labeled as a non-member. In the second step, the algorithm checks structure-reachable vertices, a specified similarity measure between vertices, from a core vertex. The third step classifies non-member vertices as *hub*s, if isolated vertices have edges connecting to two or more clusters, or as outliers. The worst case running time of the algorithm is $O(n^2)$, however it reduces to $O(n)$ if the graph is random.

Using minimum spanning trees of a network to cluster biological data is practical since edge removal divides one group of genes into two groups directly. Xu et al. [151] represent gene expression data as a minimum spanning tree (MST). Clusters are then found by three algorithms that use different objective functions to generate sub-trees. One objective is partitioning the tree into a specific number of sub-trees and minimizing the total edge distances of all sub-trees. The second objective is to minimize the distance between the center of each cluster and its objects. The third objective is similar to the second, except that a representative point is used instead of a center. The study reports that not much

information is lost using a tree representation of the data sets. They also propose a number of clustering algorithms for MST, where two of them guarantee global optimality for non-trivial objective functions. The algorithms are implemented as a computer software which is available upon request from the authors.

Community structure finding algorithms use network structure and attempt to optimize a measure called *modularity* [113]. Higher modularity values are desired. Community structure finding consists of dividing the network into groups according to certain structural information, like *betweenness* of edges, rather than similarity information normally used in traditional clustering approaches. In Newman and Girvan [113] and Girvan and Newman [49], the edges responsible for connecting many pairs of vertices, not the edges having the lower weights, are removed to find communities. With this technique, one can count how many paths proceed along each edge with the expectation that this number will be largest for intercommunity edges, the betweenness measure. The simplest example of the betweenness measure is based on the shortest paths. Communities are the sub-networks where the edges within have high density connections but the edges between have low density connections. Communities appear to have a hierarchical structure in most real world contexts [29]. For instance, people make up departments and departments make up a university, just like words make up sentences, sentences make up chapters, and chapters compose books. In that sense, community finding is similar to an HC approach. HC here is equivalent to starting with the network of interest, attempting to find the least similar connected pairs of vertices, and removing the edges between them iteratively.

Forming communities that maximize modularity is desired. For the modularity formulation, see the objective function of model (4) in the "Optimization Based Algorithms" section. Newman [112] expresses modularity in terms of eigenvectors of the modularity matrix of the network and proposed an algorithm which has a running time of $O(n^2 \log\ n)$ to divide the network into clusters. Ruan and Zhang [123] introduce a heuristic that combines spectral graph partitioning and local search to optimize modularity, and a recursive algorithm to deal with the resolution problem, that is being unable to find clusters smaller than a scale, in network community detection. The algorithm has a higher weighted matching score for protein community complex than [112]. The algorithm is also faster than [112] for networks having more than about 1,500 vertices. Clauset et al. [30] present a fast hierarchical agglomerative algorithm to detect community structure in very large networks. The algorithm has a time complexity of $O(m\ d \log\ n)$ where m is the number of edges, n is the number of vertices and d is the depth of the dendrogram. Schwarz et al. [127] use this algorithm to resolve functional organization in the rat brain. Newman [109] introduces a method of mapping weighted graphs to unweighted multigraphs, or graphs with multiple edges, to be able to use community structure finding algorithms [113] for weighted graphs. Gómez et al. [51] present a reformulation of modularity to be able to work on weighted, directed, looped networks defined from correlated data. It is also mentioned that other methods such as clique percolation [115] may be employed for a similar task with a relevant adaptation. The clique percolation method was used to find overlapping communities in yeast protein interaction data. Stone and Ayroles [132] propose an algorithm to maximize modularity that modulates weights of the edges of bi-

ological data, represented as a graph. The algorithm is applied on human and Drosophila melanogaster data, compared with a agglomerative HC and three spectral clustering algorithms using 10,000 simulated data sets. The proposed method has the highest percentage of correctly clustered objects and correctly separated objects for a specified number of clusters compared to others. The authors mentioned that Matlab code of the algorithm is freely available.

Label propagation is a recently developed method for finding community structure. It defines a community as a set of nodes such that each node has at least as many neighbors in its own community as in any other one. In the initial stage of the method, all nodes form a distinct community where each node has its own label. Then, at each time step, the nodes join with that community to which the largest fraction of their neighbors belong, by adopting the corresponding label. If there are multiple choices, a random decision is made with uniform distribution [138].

Lancichinetti and Radicchi [81] introduce a class of benchmark graphs to test the performance of two community structure algorithms. For a review of algorithmic methods to detect community structure in networks, see [110]. Fortunato [44] exposes community detection in graph thoroughly from definition of basic elements of community finding problem to the real world applications.

There are other graph based clustering approaches [64, 18]. To ease the use of graphs in solving problems, libraries such as The Boost Graph Library (BGL) for C++ and igraph[32] have been developed. The igraph library can be embedded into higher level programs or programming languages like C/C++, Python and R [32]. NetworkX [54]

is a Python-based package for complex network research. There are visualization and exploratory tools for gene clusters to be interpreted more easily. Cytoscape, and the gcExplorer [126], [125] package for R programming language are designed for such a purpose. Figure 1.8 illustrates two different layout for an expression data generated by Cytoscape. They are hierarchical and spring embedded layouts for protein-protein and protein-DNA interactions in the yeast galactose metabolism. Nodes and edges represent the proteins and the protein-protein interactions.

1.3.4 Optimization Based Algorithms

Optimization based algorithms may be more attractive to the OR community since optimization is at the heart of OR. Glover and Kochenberger [50] propose a new modeling and solution methodology for clustering that can be used for finding groups, or modules, in genomic data. Modules can be regarded as cliques of similar objects. They model the clique partitioning (CP) over nodes formulated as in (2), rather than over edges as in (1):

$$\text{Maximize} \quad \sum_{(i,j)\in E} w_{ij} x_{ij}$$

subject to

$$\begin{aligned} x_{ij} + x_{ir} - x_{jr} &\leq 1 \qquad \forall i,j,r \in V, i \neq j \neq r, \\ x_{ij} &\in \{0,1\} \quad \forall i,j \in V. \end{aligned} \tag{1.1}$$

$$\text{Maximize} \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} \sum_{k=1}^{K_{max}} x_{ik} x_{jk}$$

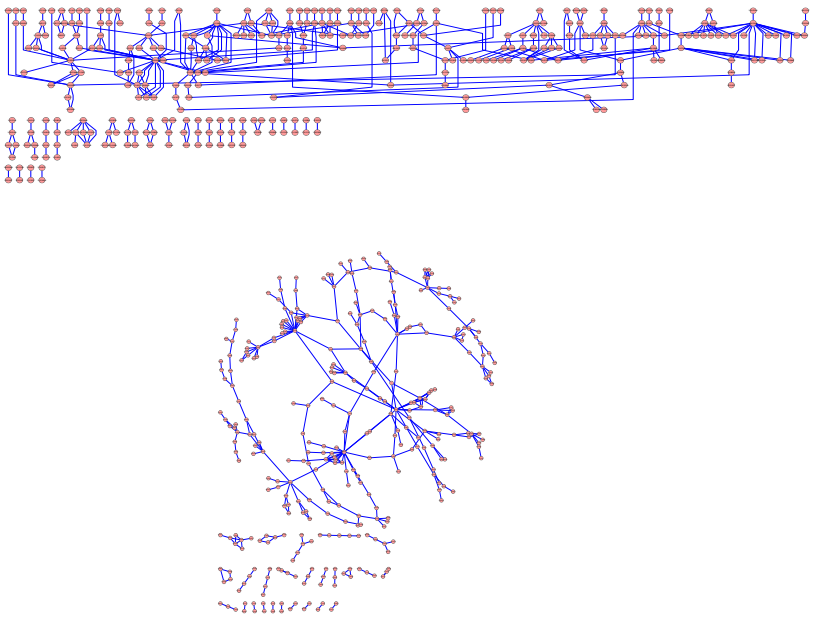

Figure 1.8

Layouts for interactions in the yeast galactose metabolism

subject to

$$\sum_{k=1}^{K_{max}} x_{ik} = 1 \qquad \forall i \in V, \tag{1.2}$$

$$x_{ik} \in \{0,1\} \quad \forall i \in V, k = 1, ..., K_{max}. \tag{1.3}$$

In the first formulation (1), x_{ij} is equal to 1 if the edge (i, j) is in the partition; 0 otherwise. The w_{ij} coefficient is the unrestricted weight of an edge between node i and node j. E and V represent the set of edges and set of vertices, respectively. In the second formulation (2), x_{ik} is equal to 1 if node i is assigned to clique k. K_{max} is the maximum number of cliques or clusters allowed, n is number of nodes, and w_{ij} is defined as in formulation (1). Formulation (2) has fewer variables and number of constraints, compared to (1). Although (2) is a quadratic model, it can be used for large instances of the CP problem. This model is similar to the one in [108] except that [50] uses the maximization objective.

Nascimento et al. [108] used a greedy randomized adaptive search procedure (GRASP) based clustering algorithm for clustering different data sets of microarrays which was guided by an integer programming model similar to (2).

Clustering based on the modularity measure introduced in "Network Based Algorithms" section uses heuristic algorithms. Maximizing the modularity measure is also used as an objective function of the integer linear program (ILP) in [19] as follows:

$$\text{Maximize} \quad \frac{1}{2m} \sum_{(i,j \in V)} (E_{ij} - \frac{deg(i)deg(j)}{2m}) x_{ij}$$

subject to

$$\begin{aligned} x_{ii} &= 1 \quad \forall i, \\ x_{ij} &= x_{ji} \quad \forall u, v, \\ x_{ij} + x_{jk} - 2x_{ik} &\leq 1 \quad \forall i, j, k \in V, \\ x_{ik} + x_{ij} - 2x_{jk} &\leq 1 \quad \forall i, j, k \in V, \\ x_{jk} + x_{ik} - 2x_{ij} &\leq 1 \quad \forall i, j, k \in V, \\ x_{ij} &\in \{0, 1\} \forall i, j. \end{aligned} \tag{1.4}$$

The decision variables x_{ij} are defined as 1 if nodes i and j are assigned to the same cluster, or 0 otherwise. E_{ij} is 1 if there is an edge between nodes i and j, 0 otherwise. $deg(i)$ and $deg(j)$ are the degrees of nodes i and j. m is the total number of edges. Equalities and inequalities are reflectivity, symmetry, and transitivity constraints. The number of variables can be reduced to $\binom{n}{2}$, and the number of constraints can be reduced to $\binom{n}{3}$ by eliminating redundant variables and constraints where n is the number of nodes. Agarwal and Kempe [1] used the same ILP model with a different variable definition. To solve their model, they use a linear programming (LP) rounding algorithm and a local search proposed by Newman [111]. LP rounding provides upper bound. Chen et al. [25] uses LP to study the community structure of networks.

Lee et al. [84] propose a graph-based relaxed optimization approach. They model clustering as a quadratic program. Their method automatically determines data distributions without a priori knowledge about the data that makes it superior to spectral clustering approach.

Tan et al. [134] propose a novel clustering approach based on mixed integer nonlinear programming (MINLP). They convert their model to mixed integer linear programming (MILP) by introducing new variables and constraints. They apply a generalized Benders' Decomposition method to obtain lower and upper bounds for the solution of MILP to converge to optimal global solution for large data sets. Their formulation is as follows:

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{c}\sum_{k=1}^{s} w_{ij}(a_{ik} - z_{jk})^2$$

subject to

$$\begin{aligned} \sum_{j=1}^{c} w_{ij} &= 1, \forall i, \\ w_{ij} &\in \{0,1\}\ \forall i,j, \quad and \quad z_{jk} \in R\ \forall j,k. \end{aligned} \tag{1.5}$$

Here, a_{ik} is the measure of distance for gene i having k features. w_{ij} are binary variables having value of 1 if gene i is in cluster j, or 0 otherwise. This model is expanded as:

$$\text{Minimize} \quad \sum_{j=1}^{c} w_{ij} \sum_{i=1}^{n}\sum_{k=1}^{s} a_{ik}^2 - \sum_{i=1}^{n}\sum_{j=1}^{c}\sum_{k=1}^{s} a_{ik} w_{ij} z_{jk} + \sum_{j=1}^{c}\sum_{k=1}^{s} z_{jk} \sum_{i=1}^{n} w_{ij}(z_{jk} - a_{ik})$$

Since the vector distance sum of all genes within a cluster to the cluster center, z_{jk}, must be 0, following optimality condition holds:

$$\sum_{i=1}^{n} w_{ij}(z_{jk} - a_{ik}) = 0, \forall j, \forall k. \tag{1.6}$$

Parameter $suit_{ij}$ is introduced to the model to restrict some genes for specific clusters. It takes a value of 1 only for the cluster in which a gene is allowed to be involved, but 0 for the other clusters. This parameter reduces the computational burden of the problem. Then, the formulation becomes:

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{k=1}^{s} a_{ik}^2 - \sum_{i=1}^{n}\sum_{j=1}^{c}\sum_{k=1}^{s}(suit_{ij})(a_{ik}w_{ij}z_{jk})$$

subject to

$$\begin{aligned}
(suit_{ij})(z_{jk}\sum_{i=1}^{n} w_{ij} - \sum_{i=1}^{n} a_{ik}w_{ij}) &= 0 \qquad \forall j,k, \\
\sum_{j=1}^{c}(suit_{ij})w_{ij} &= 1 \qquad \forall i, \\
1 \le \sum_{i=1}^{n}(suit_{ij})w_{ij} &\le n-c+1 \ \forall j, \\
w_{ij} &\in \{0,1\} \qquad \forall i,j, \\
z_{jk}^L &\le z_{jk} \le z_{jk}^U \ \forall j,k. \qquad (1.7)
\end{aligned}$$

The first set of constraints are necessary optimality conditions; the second set of constraints assure that each gene belongs to exactly one cluster. The third set of constraints assure that each cluster has at least one gene but no more than $n-c+1$ genes. The lower and upper bounds for the continuous variable z_{jk} are z_{jk}^L and z_{jk}^U. To convert this non-linear model to a linear model, new variables and constraints are added to the model:

$$y_{ijk} = w_{ij} z_{jk},$$

$$z_{jk} - z_{jk}^{U}(1 - w_{ij}) \leq y_{ijk} \leq z_{jk} - z_{jk}^{L}(1 - w_{ij}),$$

$$z_{jk}^{L} w_{ij} \leq y_{ijk} \leq z_{jk}^{U} w_{ij}, \forall i, \forall j, \forall k. \tag{1.8}$$

Tan et al. [135] apply an algorithm guided by this model to three different microarray data sets. Hayashida et al. [56] propose two graph theoretic approaches: 1) maximizing the number of genes covered by at most a constant number of *reporter gene*s, which are used to report the expression level of a gene, and 2) minimizing the number of reporter genes to cover all the nodes of the directed network. McAllister et al. [99] present a computational study to solve the distance-dependent rearrangement clustering problem by using MILP. They present three models based on the relative ordering of the elements, assignment of the elements to a final position, and distance assignment between a pair of elements. They report that their models can be used for discoveries at the molecular level. Dittrich et al. [35] deal with the problem of finding biologically meaningful sub-networks from PPI data. They transform that problem to the price-collecting Steiner tree (PCST) problem, where the total sum of the edge weights of the subtree and the profits associated with the nodes not in the subtree are minimized. They are able to solve large instances of the problem in a reasonable time to optimality by the ILP approach for the transformed problem. Melia and Pentney [100] formulate spectral clustering in a directed graph as an optimization problem with the objective of weighted directed cut in the directed graph. *Metaheuristic Clustering Algorithms* Metaheuristics and heuristics are algorithms that generate feasible solutions to

hard problems. They are used when it is impossible or too time costly to find an optimal solution to a problem. Metaheuristics are generally used in partition based clustering and are rarely used in HC [14]. Genetic algorithms (GA), ant colony optimization (ACO), Tabu Search (TS), and simulated annealing (SA) are some widely used metaheuristics.

GAs are population-based heuristics and the steps are inspired from biological phenomena. Bandyopadhyay et al. [12] use a two-stage GA to cluster one artificial and three real microarray data sets. They employ a variable string length genetic scheme and multi-objectivity. In the first stage of the algorithm, they use an iterated version of Fuzzy C-Means (FCM), which is fuzzy version of K-means to detect the number of clusters. They compare the algorithm to an HC, an SOM and a Chinese restaurant-based clustering (CRC) algorithm [117] using two cluster validation indexes: *adjusted rand index* [67] for artificial data set only because the rand index uses real clusters as input, and *silhouette index* [122]. [80] also employ a multi-objective GA. One of the objectives is minimizing the total variation within clusters, which is identical to K-means' objective. The other one is minimizing the number of clusters in a partition. Iris and Ruspini data sets are used. [41] present a Pareto-based multi-objective GA where objectives to be optimized are validation indices. Pareto set, the set including the best partitions based on different objective funtions, is used to ensemble the partition pairs to have a consensus partition. The method is applied to six microarray data sets. The method is computationally expensive, including the dissimilarity matrix calculations the complexity is $O(n^2d)$ where n is the number of objects, and d is the dimension of an object The crossover algorithm is $O(nk^2)$, where k is the number of clusters in the consensus partition. Wei and Cheng [147] develop an entropy-

based clustering method in which a GA is applied. The method uses an adaptive threshold for similarity between objects and a fitness function to calculate the clustering accuracy. It is compared with K-means, FCM, and an entropy-based fuzzy clustering method upon which the proposed algorithm was developed. Four data sets, one of which is breast cancer data, are used for comparison. [55] present a GA based biclustering algorithm with a homogeneous clustering criterion, introduce a cluster stability criterion. The method is used for metabolomics data sets. The proposed clustering routines are also available.

He and Hui [57] investigate ACO-based algorithms for clustering gene expression data. The proposed algorithm, Ant-C, consists of four phases: initialization, tour construction, pheromone update where ants leave trails on the ground to guide other ants, and cluster output. Ant-C generates a fully connected network where each node is a gene and each edge is a similarity weight, or pheromone intensity. Average pheromone intensity is used as a threshold to break the linkage of the fully connected network to form clusters. MSTs are used in case of a partially connected network to break the linkage of the network. Pheromone intensities are used as weights of the spanning tree. After finding the MST, it is partitioned into sub-trees that form the clusters. Robbins et al. [120] uses an ACO algorithm for the feature *selection problem* in gene expression data.

TS moves away from the trap of local optimality by using diversification strategies. [53] apply a TS strategy to K-harmonic means clustering to avoid being trapped at local minima. The method is tested on Iris data. SA [52, 20] also uses diversification strategy to avoid being trapped in local optima. There are many other heuristic clustering approaches for biological data. Particle swarm optimization (PSO) [161, 89, 37, 70], GRASP [34],

honey-bee mating [43], memetic algorithms [102], furthest-point-first heuristic [47] are some of them.

1.3.5 Other Algorithms and Issues

Clustering approaches are not limited to the methods listed in the sections above. The following explain some of the clustering approaches which can be classified in one or more of the above sections, or in a different section.

Fuzzy clustering allows an object to be assigned to more than one cluster. The strength of each object's belonging to a cluster is defined by a membership function that has a value between 0 and 1. The summation of membership values for each gene over all clusters is 1 [23]. For fuzzy clustering implementation on biological data, Ravi et al. [118] propose two fuzzy algorithms, variants of FCM, based on a *threshold accepting* heuristic. The algorithms are compared with FCM using *E. Coli*, Iris, and Thyroid data sets. The comparison is based upon the number of clusters and the optimal values of objective functions. Ceccarelli and Maratea [23] use a learning metric to improve FCM. The developed FCM is used on Iris, breast cancer, rat, sporulation, and yeast data sets. It is compared with FCM using a modified *entropy* index where membership values are considered as probabilities, normalized and raised to the power p. Saha and Bandyopadhyay [124] propose a GA based fuzzy method having a computational complexity of $O(k\ n \log\ n\ p\ g)$ where k is the soft estimate for upper bound of the number of cluters, p is the population size and g is the number of maximum generation. The method is compared with an information based clustering algorithm using yeast expression data set and validated using both a bi-

ological validation tool and silhouette index. [107] improve both a FCM and a GA based fuzzy clustering algorithms using a support vector machine (SVM). The method is tested on diverse microarray data sets using C-rand and silhouette indices. Alshalalfah and Al-hajj [5] also use FCM with SVM on three different microarray data sets. There are many other fuzzy clustering algorithms [61, 98, 11].

Biclustering, or subspace clustering, finds a subset of similarly expressed genes over a subset of samples. It simultaneously clusters both rows, genes, and columns, conditions or samples, of a data matrix, or gene expression matrix [104]. One justification to use biclustering is that microarray data has large number of features, or genes, which may not be relevant to the features in which a researcher is interested, and these features mask the contribution of the relevant ones [104]. Another justification is that co-expressed genes under certain conditions behave mostly independently under different conditions [34]. Li et al. [87] extend a generic biclustering approach incorporating overlapping capability. It is mentioned that the method is convenient for finding genomes with high genetic exchange and various conserved gene arrangement. The time complexity of the algorithm is $O(m^3(n^2 + log^2 m))$ where m is the number of data points and n is the number of dimensions. Subspace clustering error, row clustering error, coverage and discrepancy in the number of clusters are used for validation purpose. Christinat et al. [28] show that using discrete data coupled to a heuristic on continuous one leads to biclusters which are biologically meaningful. Li et al. [86] present a qualitative biclustering algorithm where an expression data matrix is composed of 0 and signed integer values. The algorithm is applied on *E. coli* and yeast data sets and compared with other biclustering algorithms

using biological enrichment criterion. Both the source code and the server version of the algorithm are available. Cano et al. [22] present an intelligent system for clustering. The system employs three novel algorithms. Two of them are biclustering algorithms. Madeira and Oliveira [95] and Busygin et al. [21] present comprehensive surveys of algorithms used in biclustering.

Shen et al. [130] propose a joint latent variable model for integrative clustering called iCluster. iCluster is scalable to different data types, and enables the opportunity for next generation sequencing, a new emerging technology alternative to microarrays. Ma and Chan [93] propose an iterative approach to mine overlapping patterns in gene expression data. Their approach consists of two steps. First, initial clusters are generated using any clustering algorithm. Second, cluster memberships are reassigned by a pattern discovery technique. At the end, a gene stays in the same cluster, changes clusters, or is copied to another cluster. Shaik and Yeasin [128] present a unified framework to find differentially expressed genes from microarray data. The framework consists of three modules: gene ranking, significance analysis of the genes, and validation. An adaptive subspace iteration algorithm is used for clustering in the first module. Subspace structure is identified by an optimization procedure.

Yip et al. [155] present some search algorithms to find dense regions in categorized, which are discretized, or dichotomized, gene expression data. Meng et al. [101] introduce an enrichment, a validation based on biological knowledge or database, constrained time dependent clustering algorithm. The algorithm is specially designed for time course data and integrated with biological knowledge guidance. Nueda et al. [114] also present three

novel methodologies for functional assessment of time course microarray data. Ernst et al. [40] design an algorithm specifically for clustering short time series expression data.

Model-based clustering algorithms [74, 58, 146, 150, 83] have an assumption that biological data follow a statistical distribution and try to recognize the distribution. Information-criterion based clustering algorithm[90], adaptive clustering [27], neural network [156], cluster ensemble [65], consensus clustering[105], game theoretical applications [106, 92] are some of the diverse clustering approaches.

Table 1.3 presents a summary of the reviewed algorithms, including one from each class of algorithms based on availability of the algorithm, number of times it is cited and being recent. CRC is abbreviation for Chinese Restaurant Cluster, ISA and memISA are biclustering algorithms, and CAGED is an algorithm designed for time series data. g is clique size, s is the significant profile size, e is the number of edges.

Table 1.3

Summary of Reviewed Algorithms

Class	Algorithm	Compared with	Biological Data Sets Used	Validation Method	Complexity	Availability
Flat	Richards et al. (2008)	CRC, ISA, MemISA	brain expression (~20,000 genes)	biological	O(iknm)	software
Hierarchical	Langfelder et al. (2008)	HC, PAM	Drosophila PPI	external, biological	$O(n^3)$	R package
Network	Huttenhower et al. (2007)	8 clustering algorithms	yeast (~6,000 genes)	biological	$O(n^g)$	Java implementation
Optimization	Dittrich et al. (2008)	a heuristic approach	human PPI (~2,500 proteins)	biological	$O(e^2n+en^2logn)$	software
Other	Ernst et al. (2005)	K-means, CAGED	human (50 profiles)	biological	s^4	Java implementation

1.3.6 Choice of an Algorithm

One issue in choosing a clustering approach for data is to decide about the suitability of clustering algorithms for a biological application. Andreopoulos et al. [6] address a general set of desired features that change based on application and data type used: scalability, robustness, order insensitivity, minimum user-specified input, mixed data types, arbitrary-shaped clusters, and point proportion admissibility. Scalability is concerned with time and memory requirements, which increase as the data set becomes larger. They define robustness as ability to detect outliers. Order insensitivity means that clusters are not changed as the objects' order changes. Minimum user-specified input, as the name suggests, emphasizes a clustering algorithm's reliance on user-specified input as little as possible. Mixed data types and arbitrary shaped clusters refer to allowing objects to have numerical descriptive attributes and an algorithm's ability to find arbitrarily shaped clusters. Point proportion admissibility means stability of the results when objects are duplicated and re-clustered.

Another issue for choosing a clustering approach is the performance evaluation of the approach. Internal and external performance measures are developed for evaluation. Internal measures rely on the structure of the partition, whereas external measures use external information, such as the knowledge of the real clusters. Real clusters for samples are known in advance, since samples are the designed experiments or the time course data. Clusters of genes are not known in advance except for the well annotated genes. Thus, using external performance measures for algorithms that cluster genes is hard. After clustering genes, researchers validate the clusters from gene databases if specific knowledge

about the genes is available. Modularity, discussed in the "Network Based Algorithms" section is an internal measure that makes use of the network's structure. Modularity is a strong measure in the sense that biological networks exhibit some common structures. Silhouette [122] is another internal measure based on the compactness and separation of the clusters. For an application of silhouette index, see [12]. *adjusted rand index*, or C-rand [67], is an external measure of agreement between two different partitions, one of which is real. C-rand is applicable even if the partitions do not have the same partition size [152]. Yeung et al. [152] give an example of calculating the C-rand value. For other performance measures see [91], and [156]. Using simulated data, clusters' stability on a partition [42], reproducibility of the clusters [46], statistical significance between clusters [160], and comparing clustering of a combination of conditions with remaining conditions [153] are other ways to test the performance of a gene clustering algorithm.

1.4 Conclusion and Future Research for the Operations Research Community

Clustering is fundamentally an optimization problem [7]. The clustering problem has awakened more interest in the statistics and computer science disciplines than in the optimization community [136]. Hence, the OR community, with an optimization paradigm, may become involved in and contribute more to clustering problems in the bioinformatics, computational, and systems biology disciplines.

No clustering algorithm exists with the best performance for all clustering problems. This fact makes it necessary to use or design algorithms specialized for the task at hand. Algorithmic methods are challenged by the introduction of high-throughput technologies

[15]. Guiding any clustering method with biological theory regarding genomic data is essential. Mathematical programming (MP) formalism offers flexibility to incorporate biological knowledge, and it is crucial to use algorithms guided by MP models for genomic data analysis [7]. Hence, IP models taking into account the biological knowledge would be a promising research direction. Clustering of genomic data as a data mining problem includes challenging problems providing a relatively hot and fruitful arena for the OR community [50]. OR has been an underutilized resource in the research agenda popularized by network science [3]. Network-based clustering problems may involve more OR researchers to contribute the agenda.

1.5 Glossary

Activator: a metabolite that regulates genes by increasing the rate of transcription.

Adjusted rand index: see index.

Betweenness: here defined for an edge. The number of shortest paths proceeding along an edge.

Biological database: database used for validating whether a clustering algorithm generates clusters that are biologically meaningful. Gene ontology (GO) is one of the most widely used biological database.

Classification: a supervised learning technique assigning objects into the groups already known.

Cluster: is a group that includes objects with similar attributes. Clustering is an unsupervised learning technique. Output of a clustering is a set of clusters including similar objects, i.e., genes. Clustering is also an exploratory technique for network decomposition [85]. Clustering gathers objects into the same group based on a cluster definition or criterion.

Clustering: see cluster.

Connectivity: minimum set of genes required to inhibit the synthesis of a product.

C-rand: see index.

Data pre-processing: a process applied to raw expression data obtained from microarray experiment. Pre-processing includes *quality assessment*, filtering, normalization also referred as low-level analysis.

Dendrogram: a tree showing the hierarchical relations between groups of objects. Level of a dendrogram is the cut-off value to cut the dendrogram to obtain the clusters.

Distance measure: a measure of relationship between a pair of objects. Eucledian(e_{ab}), Manhattan (m_{ab}), Minkovski (mn_{ab}) are some examples. Correlation (c_{ab}) is also a widely used distance measure. However, $\sqrt{1 - c_{ab}}$ approximation is used to satisfy the triangle inequality attribute of a metric.

$e_{ab} = \sqrt{\sum_{i=1}^{n} (d_{ai} - d_{bi})^2}$, $m_{ab} = \sum_{i=1}^{n} (d_{ai} - d_{bi})$, $mn_{ab} = \sqrt[p]{\sum_{i=1}^{n} (d_{ai} - d_{bi})^p}$

Entropy index: see index.

Eucledian distance: see distance measure.

eQTL: expression quantitative trait loci, genomic locations where genotype affects gene expression.

Expression pattern: pattern that a gene exhibits through different conditions, i.e., samples.

Factor graph: spanning sub-graph of a graph.

Feature: attribute of a microarray either referrring to a spot of it or a gene.

Feature selection problem: selection of the most important, relevant genes for further analysis to reduce the dimensions of high dimensional data.

Filtering: removing the genes that don't exhibit significant expression change through conditions or the genes, expression of which are below a certain threshold.

Gene: a functional unit of DNA with coded information. Reporter genes encode fluorescent proteins by which the expression level of gene can be observed [56]. The study of

genes is called genomics. Genome refers to all of the fundamental genetic units, hereditary information, in a biological cell.

Gene expression: transcription of DNA into RNA.

Genome: see gene.

Genomics: see gene.

Hub: gene with high connectivity.

Index: measure for validating the performance of a clustering algorithm. *Adjusted rand index* for partitions P_1 and P_2 (C-$rand(P_1, P_2)$), as an external validation index, is one of the most widely used index for comparing the partition generated by a clustering algorithm with the real partition. Silhouette index for partition P_1 ($S(P_1)$), as an internal validation index, is used when the real partition of a biological data is not known. Partition entropy index (PE) is a measure of asymmetry. C-$rand(P_1, P_2)$, $S(P_1)$ and PE formulations are:

C-$rand(P_1, P_2)$ = $\frac{\sum_{i,j}\binom{n_{i,j}}{2} - [\sum_i \binom{n_{i\cdot}}{2} \sum_j \binom{n_{\cdot j}}{2}]/\binom{n}{2}}{1/2[\sum_i \binom{n_{i\cdot}}{2} + \sum_j \binom{n_{\cdot j}}{2}] - [\sum_i \binom{n_{i\cdot}}{2} \sum_j \binom{n_{\cdot j}}{2}]/\binom{n}{2}}$ where $n_{i,j}$ is the number of objects at the intersection of clusters i and j, i is the cluster index for P_1, j is the cluster index for P_2. $n_{i\cdot}$ is the number of objects in cluster i.

$S(P_1, P_2)$ = $\frac{\sum_{i=1}^{n} \frac{g(i)-a(i)}{max(o(i),s(i))}}{n}$ where n is the number of genes, $o(i)$ is the minimum of average distaces from gene i to the genes in the other clusters. $s(i)$ is the average distance from gene(i) to the remaining genes in the same cluster.

PE = $\frac{1}{n}\sum_i^n \sum_j^k \mu_{ij} log_a \mu_{ij}$ where k is the number of clusters and μ_{ij} is the membership of i in j [23].

Manhattan distance:see distance measure.

Metabolite: product of metabolism.

Microarray: a chip consisting of thousands of microscopic spots, i.e, features containing genes. Two signed microarray data includes both positive and negative values corrresponding to up and down regulation respectively.

miRNA: small RNA that binds to mRNA to regulate expression.

mRNA: the RNA transcribed by a gene to be translated into a protein [97].

Modularity: a measure of improvement on random connectivity.

Next generation sequencing: a high throughput technology that allows measuring DNA sequences directly rather than indirect way of measuring, i.e., image processing of microarrays.

Noise: irregularities in the expression data. The sources of noise are sample preparation and hybridization process [143]. Genes that are irrelevant to clustering, i.e., non-informative genes [72] are also regarded as noise.

Normalization: transformation of raw expression data to ensure the comparability of gene expression levels across samples with the purpose of minimizing the systematic variations arising from technological issues [133].

Object: gene or sample.

Partition: the output of a clustering algorithm, the set of the clusters generated.

Priority queue: a heap data structure. A binary tree has a heap property if and only if it is empty or the key of the root has a higher value than all of its and subtrees of the tree has a heap property as well. The root node has the highest value and once it is extracted, regeneration of a single tree from two subtrees takes $O(logn)$ time where n is the number of nodes. Heap tree is filled from left to right, once the root is deleted the right most leaf is

taken as the root. Figure 1.9 illustrates a tree with heap property: a) First, tree extracts the root and then the first move is bringing the right most leaf to vacant root position. Second, root value, i.e., 6 is swapped with left subtree's root value, i.e., 8 and the resulting new heap tree is shown as in b). The number of swaps is at must the length of the complete binary tree which is $logn$.

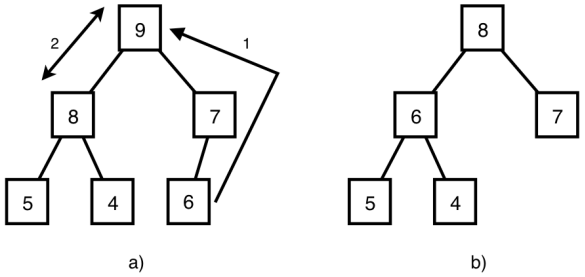

Figure 1.9

Priority queue

Quality assessment: a procedure to be applied on microarray data to ensure that the data is ready for further analysis.

Regulatory site: 5-15 base-pairs of genes.

Reporter gene: see gene.

Repressor: a protein that represses the transcription of genes.

Reverse engineering: also referred as deconvolution, process of analyzing biological data to infer about the interaction of biological components.

Sample: each microarray chip.

Scale-free topology: a network topology where the degree distribution of nodes follow a power law.

Silhouette index: see index.

Small world property: a network where each node has a small number of neighbor but can reach to other nodes at a small number of steps.

Systems biology: a discipline that deals with the computational reconstruction of biological systems.

Transcription factor (TF): activator or *repressor* proteins produced by genes.

Threshold accepting: a local search strategy that allows up-hill moves for a minimization objective.

Two-signed microarray expression data: see microarray.

Validation: assessing the performance of a clustering algorithm either using performance indices or biologically.

CHAPTER 2

A NEW MIMIMUM SPANNING TREE BASED HEURISTIC

Biological data may be represented by networks. For example, gene expression data may be regarded as a complete network where the genes are nodes of the network, edges are relations between genes and pairwise correlation values obtained from expression data are the strength of the relation, edge weights, of the gene pairs.

Clustering network data is a graph partition problem which has many variations such as clique partition and K-way equipartition. This partitions the vertices of a graph into k sets of equal size to minimize the weight of the edges within each set [71]. Since the graph partitioning problem is NP-hard [8], efficient heuristics to find meaningful solutions are developed [78].

A minimum spanning tree of a graph includes all of graph's vertices. Using minimum spanning trees (MSTs) of a network to cluster biological data is practical since edge removal divides one group of genes into two groups directly. Removing $n-1$ edges from a tree divides the nodes into n different groups. Xu et al. [151] demonstrate that no essential information is lost with an MST representation for clustering purposes. Moreover, an MST representation may overcome the computational burden of graph based calculations and difficulties with dependency on the geometrical shapes of the clusters [151]. Determining the edges to remove and developing a quality measure or objective function, for

evaluating the clusters are the most important aspects to develop a MST based heuristic. It is desirable to have both tight and separated clusters since tight and isolated clusters are more likely to have interdependent relationship. However, one usually either seeks to maximize similarity within clusters or distance between clusters.

A new objective is proposed that seeks to obtain tight and separated clusters at the same time. The objective function assumes a binary graph where there is a relation or not. The idea is that clusters should have as small diameters as possible while an object of a cluster should have as small number of connections with other clusters as possible. In order to achieve this objective, the most central or between edges of the MST are removed iteratively. The betweenness of an edge is the number of times an edge appears on shortest paths between any two node pairs. The betweenness calculation of the edges is described in [113]. The shortest path betweenness for use in the heuristic is adopted.

The work flow starts with Pearson correlation calculations compared upon expression data sets. Correlation values are used as edge weights to construct the gene co-expression network. The weighted network is transformed to a binary network using a threshold retaining the strongest edges while ensuring the network is still connected such that removal of one more edge makes the network disconnected. TSI values are calculated using the partition and the binary network. In addition to correlation calculations, expression data are also used to calculate 6 different distances: Eucledian, Chebyshev, Manhattan, Canberra, Minkovski, and 1-Pearson correlation. K-means, PAM and B-MST use these distance measures and the given number of clusters to generate partitions. Partitions are

also used to calculate adjusted rand index values. The work flow is shown in Figure 2.1 and an example is provided in section 2.

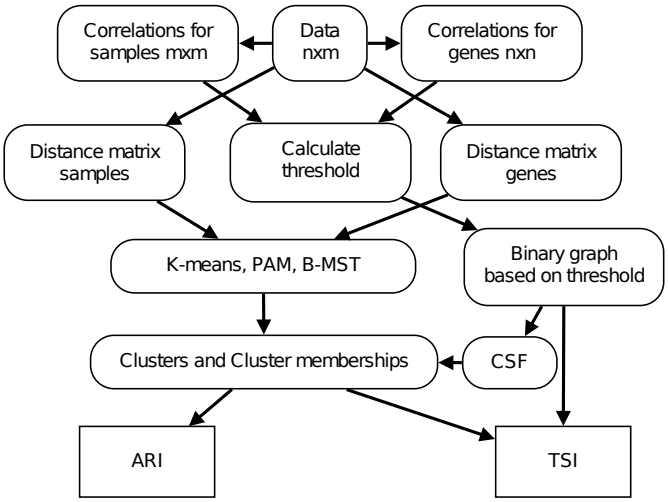

Figure 2.1

Flow of work

The chapter is organized as follows: the second section describes the B-MST method in detail, the third section describes the comparison methods and test data sets, the fourth section presents both external and biological validation results, and the fifth section is the discussion and conclusion.

2.1 The B-MST Approach for Clustering

The B-MST heruistic has two phases. First, an initial solution is generated by finding an MST of the expression data and the corresponding TSI value is calculated. Second, a local search mechanism is introduced to improve the TSI value. The algorithm is imple-

mented in R and the igraph library [32] is used for applying Prim's algorithm to generate MSTs and other graph operations.

Figure 2.2 summarizes how the initial solution is generated using B-MST. An MST is generated using distance values between gene pairs as edge weights of the co-expression network. $n-1$ edges are removed from the MST to obtain n clusters. Betweenness values of the edges are used to decide which edges to remove. The edge with the highest betweenness value is removed and all betweenness values are recalculated to remove the next edge with the highest betweenness. Edge removal goes on until the desired number of clusters are obtained. For the example illustrated in 2.2, the number of clusters is chosen 2. Euclidean distance measure is used to form the MST. The smallest indexed edge is removed when there is more than one highest betweenness score. In the example, the edge, (1,3), has the smallest index.

The expression network is transformed to a binary graph using a threshold as explained in section 1. For the example graph in Figure 2.2, this threshold is 34 percent below which the binary graph is not connected. In other words strongest 5 edges are retained and removal of one more edge makes the graph disconnected. Edge weights are Pearson correlation values between gene pairs. The higher the value, the stronger the edge is. This binary graph is then used to calculate the TSI value.

Figure 2.2 a) is a representative complete expression graph with 6 nodes and 15 edges. The 6×6 Expression data was generated using 6 normal distributions with different standard deviations. 10 samples were generated by each of the normal distribution. Red dashed edges form the MST of the graph. b) is the MST of the graph in a). Red dashed edges

are the ones with the highest betweenness score that is 8. c) is the partition with two clusters. The corresponding partition vector is also shown below the clusters. d) is the binary network transformed from a).

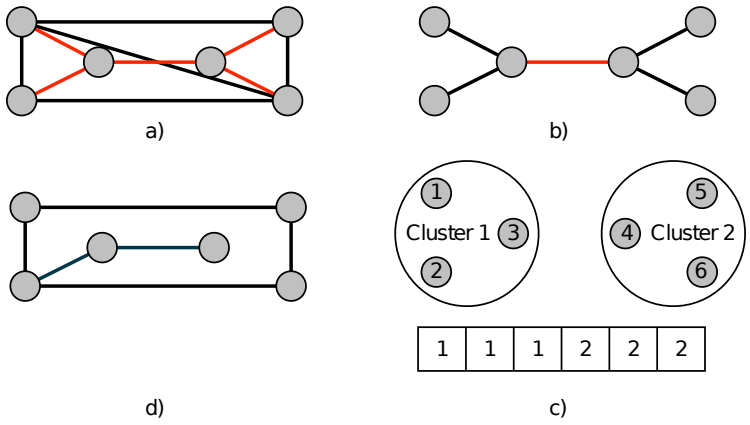

Figure 2.2

Initial Solution by B-MST

Although MSTs were used in clustering biological data [151], and the betweenness approach was applied on graph partition [113], the betweenness approach was not applied on an MST for clustering biological data.

2.1.1 Tightness and Separation Index

A new objective function, TSI, is defined and used in the heuristic. The TSI considers both the tightness and the separation of the clusters. Tightness is obtained by minimizing the maximum diameter among the diameters of clusters. The diameter of a cluster is defined as the maximum of the shortest path distances between gene pairs. Separation is

obtained by minimizing the maximum number of connections of a gene inside a cluster with other clusters. The TSI value calculation is realized on the binary graph. The shortest paths between nodes are used as distance values between gene pairs. The idea of using shortest paths is based on the transitive gene expression approach assuming that functions are often the result of many genes interacting with each other rather than a result of a simple pairwise relation[164]. However, transitive expression implies that there is at least one path, not necessarily of length 1 (assuming a binary graph) as in a pairwise relation, between two genes. The length of this path is the shortest path distance between these genes. Researchers propose that a transitive co-expression analysis applying a shortest path distance between two genes as in (Figure 1.5) gives more biologically meaningful results, rather than a direct pairwise distance measure [162, 164]. The TSI is formulated as follows:

$$D_{max} + k_{max}^{out} \tag{2.1}$$

$D_{max} = \max_{s \in S}\{D_s\}$, $S = \{1, 2, ..., c\}$ where c is the number of clusters. $D_s = \max_{i,j \in N, i \neq j}\{d_{ij}\}$, $N = \{1, 2, ..., n\}$ where n is the number of genes, d_{ij} is the shortest path distance between gene i and gene j. $k_{max}^{out} = \max_{i \in N, i \neq j}\{\sum_{j=1}^{n} a_{ij} - \sum_{j=1}^{n} a_{ij}x_{ij}\}$, where a_{ij} is 1 if genes i and j are connected, 0 otherwise and x_{ij} is 1 if i, j are in the same cluster, 0 otherwise.

For example, TSI value for the partition in Figure 2.2 c) is 5 where D_{max} is 4 and k_{max}^{out} is 2.

2.1.2 Local Search

Local search seeks for improvement on objective function value based on a neighborhood definition. Here, neigborhood is defined in such a way that a partition P' is a neighbor to a partition P if a gene in P is transferred from its current cluster to another cluster with which it has a connection. Starting from the first gene of candidate genes list (Clist), which includes the genes that have at least one connection with other clusters, a gene is transferred to the cluster with which it has the highest number of connections. This is the first cluster in the transfer list (Tlist(i)) that includes the clusters to which gene i has at least one connection, and this list is sorted descending order of the number of connections that the gene i has with other clusters. New objective value is calculated. If the new value is smaller than the initial objective function value, the partition, objective value, and transfer list are updated. This procedure is repeated until there is no improvement and n number of steps have been executed after an improvement, where n is the number of genes. The local search procedure is shown in Figure 2.3. The second and the fourth objects are transferred to the first cluster. TSI value changes from 6 to 4.

Local search transfers the nodes to the clusters which they have the highest number of connections, if this transfer would improve the objective function value. For example, the cluster membership changes with applying local search for the partition in Figure 2.2.

It takes $O(cn^2)$ time to find the initial solution where c is the number of clusters, using B-MST. This is due to betweenness calculations taking $O(n^2)$ time and are repeated $c-1$ times. Local search takes $O(cn(m+n))$ time to find the best neighboring solution for a given solution, where m is the number of edges in the binary graph.

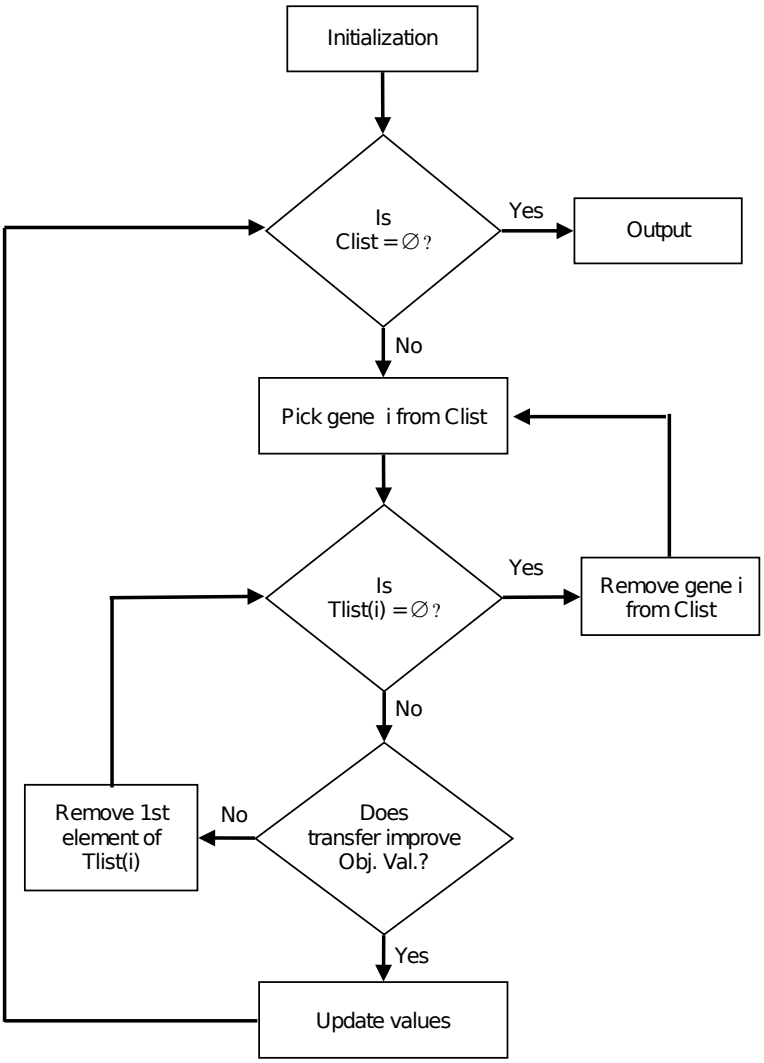

Figure 2.3

Local search procedure

2.2 Comparison Methods and Data Sets

Here, the performance of B-MST method is compared to K-means, PAM and CSF. K-means is implemented in the R base package, PAM is implemented in R *cluster* package, and the community structure finding algorithm [111] CSF is implemented in the R *igraph* package. One reason for choosing K-means and PAM is that they are widely used, and fast in clustering high dimensional data. The CSF is a recent, fast and well cited method. K-means has a time complexity of $O(tcnm)$ [96] where t, c, n, m are the number of iterations, clusters, objects, and attributes respectively. PAM takes $O(c(n-c)^2)$ for each change and CSF is $O(n^3)$. The system times for B-MST, K-means, PAM and the local search on Leukemia data set using Euclidean distance measure are 0.086, 0.128, 0.131, 4.355 respectively.

12 datasets are used for external validation and 2 data sets are used for biological validation. The features of the data sets are summarized in Table 2.1.

The microarray is a device which measures expression (abundance of RNAs) of thousands of genes simultaneously. BreastA and BreastB are cancer diagnosis microarray data sets having 98 and 49 samples respectively with 1213 attributes. BreastA is generated using 2-channel oligonucleotide microarrays and BreastB is generated using 1-channel microarray technology. DLBCLA is a diffuse large B-cell lymphoma data set having 141 samples with 661 attributes. Tumor specimens and retrospective clinical data from 176 DLBCL patients were analyzed and 80 percent of the samples (141/176 tumors) were used. The protein data set has 698 protein folds with 125 attributes. MultiA is a gene expression data set with 103 cancer type samples and 5565 genes. Novartis is the same data set which

Table 2.1

Summary of Data Sets

Data sets	# of objects	# of features	# of classes
ALB	38	722	3
Brain	37	781	5
cGCM	90	630	13
Leukemia	248	985	6
LungA	197	188	4
Novartis	103	502	4
BreastA	98	1213	3
BreastB	49	1213	4
DLBCLA	141	661	3
Protein	698	125	4,27
CNS	112	9	4
Yeast1	384	17	5
Yeast2	474	7	NA
Yeast3	2467	79	NA

has been normalized and the number of genes reduced to 1000. BreastA, BreastB, DLBCLA, DLBCLB and MultiA are pre-processed by [63]. The data sets mentioned till here are described and addressed in [108]. The ALB, Leukemia, Brain, cGCM, LungA cancer data sets are obtained from http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi. Yeast1 is the yeast cell cycle data described in [154]. CNS rat data and Yeast2 yeast sporulation data are addressed in [12]. Yeast3 is the yeast cell cycle data mentioned in [39].

All of the data sets except the last two yeast data are used for externally validation. Adjusted rand index (ARI) [67] is used for this validation. Higher ARI values indicate that partitions generated are closer to the real ones. ARI values can take on between -1 and 1. The Yeast2 and Yeast3 data sets are used for biological validation. Since high ARI values correspond to low TSI values in most of the comparisons, the algorithms resulting

in the best two TSI values are compared. B-MST is compared with CSF using Yeast2 and with PAM using Yeast3 since the CSF found the best TSI value for Yeast2 and PAM found the second best (after B-MST) TSI value for Yeast3. Algorithms are compared based on significantly clustered genes with the same biological process information which is determined by Gene Ontology (GO) terms. A similar biological inference strategy that was used by [12] is employed. This strategy results in multiple selectivity values. The highest selectivity values of all the clusters obtained by each algorithm are compared.

2.3 External and Biological Validation Results

As mentioned earlier, B-MST, K-means and PAM use distance measures to generate partitions. However, the CSF is independent of a distance measure. The CSF algorithm uses the binary network to generate clusters. For B-MST, if local search does not result in better ARI value, then the initial solution and the corresponding TSI value is shown through tables 2-7. Biological inference is realized using FatiGO [2]. FatiGO reports the percentage of annotated genes for a biological process in a cluster and the same percentage for the remaining genes of the data set. Using these percentage values, selectivity values for all clusters are calculated. For a given cluster and biological process, the selectivity is the difference between the percentage of annotated genes in the cluster and the percentage of annotated genes outside this cluster for the same biological process. Highest selectivity values are compared for the CSF and PAM. The CSF was not eligible for the Yeast3 since it finds at most 14 clusters while I determined the number of clusters 15.

2.3.1 External validation

To conduct our external validation, the ARI values were found for the partitions generated by B-MST, K-means, and PAM. The first 12 data sets shown in Table 2.1 are used for the external validation. Each table presents ARI and TSI values for a different distance measure. The first column gives the names of the data sets, from the second column to the fourth column, ARI values for B-MST, K-means and PAM are given, from the fifth to the last column, TSI values for B-MST, K-means and PAM are given. The highest ARI and TSI values for each data set are in shown in bold.

Table 2.2

ARI and Objective Values for Euclidean Distance Measure

	ARI			TSI		
Data Sets	**B-MST**	**K-means**	**PAM**	**B-MST**	**K-means**	**PAM**
ALB	**0.781**	0.138	0.394	**16**	25	20
Brain	0.596	0.429	**0.774**	20	23	**19**
cGCM	**0.636**	0.115	0.228	39	39	**30**
Leukemia	0.527	0.684	**0.939**	108	101	**81**
LungA	0.069	0.765	**0.872**	**38**	40	40
Novartis	**0.946**	0.875	0.897	**34**	47	36
BreastA	0.565	**0.597**	0.527	40	**31**	31
BreastB	0.155	0.128	**0.213**	42	42	**42**
DLBCLA	0.162	0.076	**0.176**	**104**	114	108
Protein-4	0.106	**0.320**	0.203	437	**361**	428
Protein-27	0.094	**0.137**	0.090	535	518	**516**
CNS	0.085	0.030	**0.125**	**58**	60	61
Yeast1	**0.114**	0.008	0.069	**173**	186	174

Table 2.3

ARI and Objective Values for Chebyshev Distance Measure

	ARI			TSI		
Data Sets	**B-MST**	**K-means**	**PAM**	**B-MST**	**K-means**	**PAM**
ALB	**0.577**	0.311	0.241	**19**	19	20
Brain	0.503	**0.623**	0.390	23	24	**20**
cGCM	**0.636**	0.115	0.228	39	34	**32**
Leukemia	0.311	0.280	**0.425**	**104**	106	115
LungA	0.144	0.288	**0.290**	44	45	**41**
Novartis	0.491	0.270	**0.696**	**45**	50	51
BreastA	0.215	**0.269**	0.223	**35**	37	38
BreastB	0.095	**0.218**	0.050	**42**	43	42
DLBCLA	**0.117**	0.083	0.114	110	112	**108**
Protein-4	0.099	0.265	**0.308**	**433**	476	455
Protein-27	0.067	**0.133**	0.080	**533**	535	526
CNS	**0.111**	0.016	0.110	57	61	**51**
Yeast1	**0.104**	0.022	0.091	185	182	**174**

Table 2.4

ARI and Objective Values for Manhattan Distance Measure

	ARI			TSI		
Data Sets	**B-MST**	**K-means**	**PAM**	**B-MST**	**K-means**	**PAM**
ALB	**0.781**	0.092	0.394	**16**	22	20
Brain	0.662	0.420	**0.822**	19	23	**18**
cGCM	**0.553**	0.230	0.301	37	30	**29**
Leukemia	0.557	0.790	**0.947**	97	96	**80**
LungA	0.074	0.497	**0.873**	**36**	43	40
Novartis	0.897	0.555	**0.947**	35	48	**34**
BreastA	0.406	**0.597**	0.527	36	43	**33**
BreastB	**0.194**	0.128	0.129	42	42	**31**
DLBCLA	**0.402**	0.124	0.352	108	107	**102**
Protein-4	0.111	**0.314**	0.160	489	**439**	452
Protein-27	0.075	**0.125**	0.071	532	541	**527**
CNS	**0.101**	0.032	0.095	**57**	61	61
Yeast1	**0.100**	0.006	0.073	**166**	182	174

Table 2.5

ARI and Objective Values for Canberra Distance Measure

	ARI			TSI		
Data Sets	**B-MST**	**K-means**	**PAM**	**B-MST**	**K-means**	**PAM**
Leukemia	**0.547**	-0.005	0.453	**90**	115	121
LungA	**0.064**	-0.023	0.043	43	**28**	47
BreastB	0.158	**0.241**	0.129	40	42	**31**
DLBCLA	0.535	0.336	**0.697**	**100**	105	105
Protein-4	0.112	**0.272**	0.144	487	461	**349**
Protein-27	0.098	**0.149**	0.121	530	530	**523**
CNS	**0.076**	0.014	0.031	**53**	54	55
Yeast1	**0.107**	0.004	0.048	**165**	182	174

Table 2.6

ARI and Objective Values for Minkovski (P = 3) Distance Measure

	ARI			TSI		
Data Sets	**B-MST**	**K-means**	**PAM**	**B-MST**	**K-means**	**PAM**
ALB	**0.833**	0.138	0.355	**17**	21	21
Brain	0.596	0.438	**0.774**	20	22	**19**
cGCM	**0.641**	0.153	0.256	39	**31**	34
Leukemia	0.626	**0.808**	0.804	105	94	**81**
LungA	0.083	0.531	**0.887**	40	41	**40**
Novartis	0.684	0.681	**0.973**	42	38	**34**
BreastA	**0.692**	0.633	0.462	39	32	**30**
BreastB	**0.345**	0.286	0.218	**39**	42	41
DLBCLA	**0.270**	0.051	0.143	**106**	116	108
Protein-4	0.062	**0.297**	0.249	465	**454**	467
Protein-27	0.080	**0.119**	0.096	533	531	**519**
CNS	0.057	0.025	**0.083**	55	61	**53**
Yeast1	0.061	0.009	**0.065**	**166**	186	173

Table 2.7

ARI and Objective Values for Correlation Distance Measure

	ARI			TSI		
Data Sets	**B-MST**	**K-means**	**PAM**	**B-MST**	**K-means**	**PAM**
ALB	0.341	**1.000**	0.910	21	**17**	17
Brain	0.542	0.557	**0.789**	25	20	**20**
cGCM	0.527	0.554	**0.578**	41	30	**29**
Leukemia	0.594	0.575	**0.940**	108	106	**81**
LungA	0.088	**0.333**	0.317	48	41	**40**
Novartis	0.898	0.620	**0.946**	34	44	**34**
BreastA	0.406	0.470	**0.527**	**39**	42	41
BreastB	0.205	0.266	**0.420**	**41**	42	42
DLBCLA	**0.330**	0.192	0.214	103	**92**	111
Protein-4	0.099	**0.243**	0.238	432	**362**	448
Protein-27	0.089	**0.131**	0.117	535	528	**519**
CNS	0.066	**0.160**	0.134	53	**50**	51
Yeast1	0.283	**0.522**	0.445	169	**140**	173

Investigating these tables, B-MST outperformed both K-means and PAM in 6 data sets out of 12. These data sets are BreastB, DLBCLA, ALB, cGCM, Yeast1, and CNS. For example, B-MST finds the best rand index values, 0.781, 0.577, 0.781, 0.833 in 4 distance measures, Euclidean, Chebyshev, Manhattan, Minkovski and the worst value, 0.341 only once in Pearson for ALB. Remaining data sets are evaluated in a similar manner. K-means and PAM outperformed B-MST in 2 and 4 data sets respectively.

For the same 12 data sets, ARI and TSI values were also found using the CSF. As can be seen from Table 2.8, B-MST's highest ARI values are compared to the ones found by CSF. B-MST found higher ARI values for all of the data sets except Leukemia.

Table 2.8

ARI and Objective Values for B-MST and CSF

	ARI		TSI	
Data Sets	**B-MST**	**CSF**	**B-MST**	**CSF**
ALB	**0.833**	0.109	**17**	18
Brain	**0.662**	0.326	**19**	23
cGCM	**0.641**	0.364(11 clusters)	39	**29**
Leukemia	0.626	**0.661**	105	**86**
LungA	**0.144**	0.059	44	**27**
Novartis	**0.946**	0.795	34	**32**
BreastA	**0.692**	0.521	39	**35**
BreastB	**0.345**	0.238(3 clusters)	**39**	47
DLBCLA	**0.592**	0.274	**97**	105
Protein-4	**0.139**	0.137	**432**	443
Protein-27	**0.098**	0.061	530	**460**
CNS	**0.111**	0.024	57	**47**
Yeast1	**0.283**	0.281	169	**110**

From Tables 2.2-2.7, it is observed that the maximum ARI values correspond to the minimum of the TSI values in most cases. Hence, it is proposed that the partition with smaller TSI value is expected to have more biologically relevant clusters.

2.3.2 Biological Inference

GO biological process terms of the clusters are investigated using the Yeast2 and Yeast3 data sets. Yeast2 clusters found by B-MST are compared with CSF's and Yeast3 clusters from B-MST are compared with PAM's. The highest selectivity values in a cluster are chosen for comparison. The number of clusters is determined to be 8 for Yeast2 and 15 for Yeast3. The numbers are decided by visualization of the dendrograms generated by hierarchical clustering (HC) with average linkage such that clusters include enough num-

ber of genes visually. HC cut-off levels are 650 and 6 for Yeast2 and Yeast3 respectively. Clusters having less then 10 genes are not considered. The number of clusters are also supported by Dynamic Tree Cut algorithm [82]. Dynamic Tree Cut detects the number of clusters based on the shape of a dendrogram. It has user defined parameters such as minimum cluster size and the cut height of the tree. These parameters are set to reasonable values 10, 150.5 and 10, 13 for Yeast2 and Yeast3 respectively. Dendrograms for Yeast2 and Yeast3 are given in Figures 2.4 and 2.5. The Euclidean distance measure was used in all algorithms.

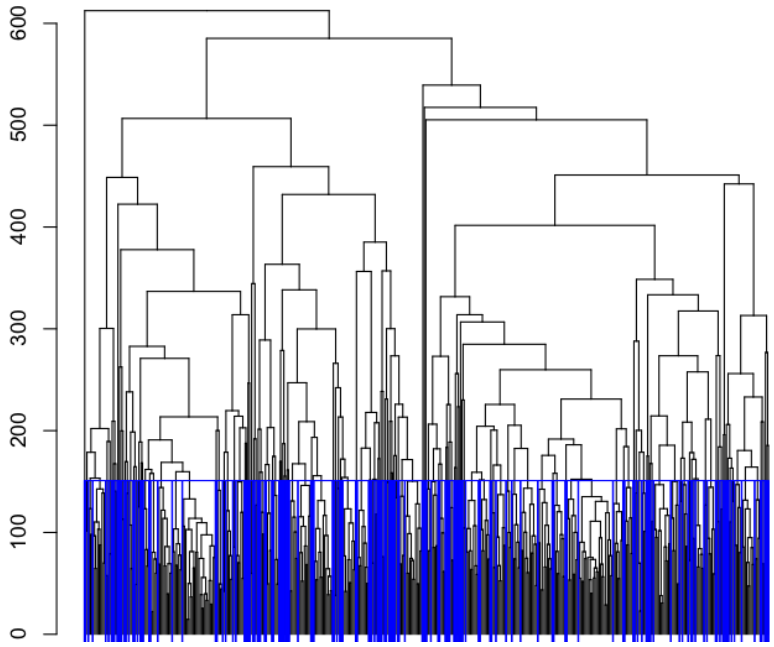

Figure 2.4

Dendrogram for Yeast2

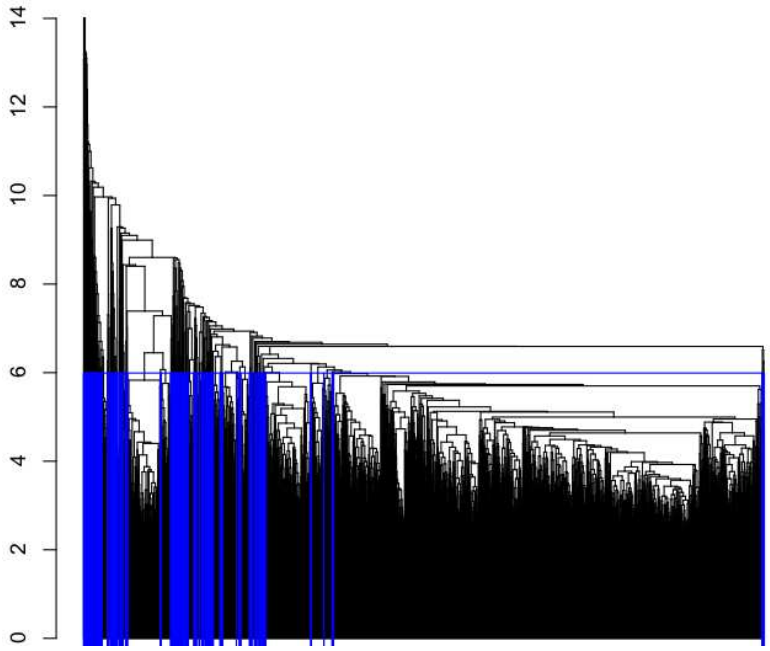

Figure 2.5

Dendrogram for Yeast3

GO terms are too general at lower levels and too specific at upper levels. Hence, GO levels are chosen between 7 and 9. The highest selectivity values of the clusters are plotted in Figure 2.6 and Figure 2.7. For example, for cluster 4 of Figure 2.6, the highest selectivity value for B-MST is 20.46 while it is 11.66 for the CSF. Zero values in the figures indicate that the there is no significant biological process found among the genes in this cluster. Negative value in the Figure 2.7 indicates that the percentage of annotated genes in this cluster for a specific biological process is less than the percentage of the annotated genes in the remaining clusters for the same biological process. In this sense, a negative value is not worse than a zero value, since it at least indicates a relationship regarding a biological process.

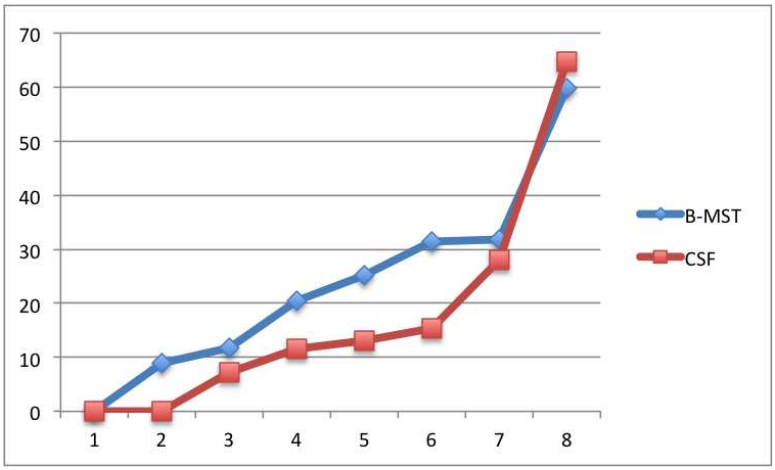

Figure 2.6

Highest selectivity values found by B-MST and CSF

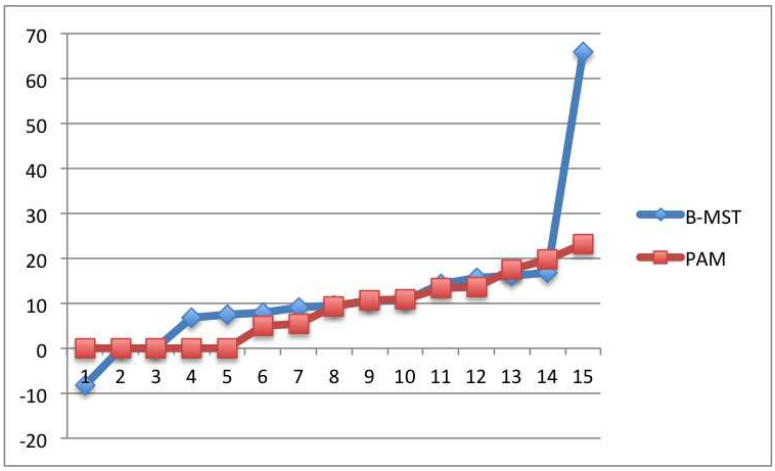

Figure 2.7

Highest selectivity values found by B-MST and PAM

2.4 Discussion and Conclusion

Clustering high throughput biological data efficiently is essential especially when there is a lack of prior information about the interactions between biological molecules. The high dimensional nature of the abundant data makes it necessary to design efficient and effective algorithms generating biologically meaningful clusters.

In this study, a minimum spanning tree based algorithm, B-MST, is developed to cluster gene expression data efficiently. The algorithm uses a new objective function, TSI, which is used as a measure of tightness and separation at the same time considering transitive distances on a binary graph to generate biologically meaningful clusters.

Many distance measures and diverse data sets were employed for ARI calculations to show that B-MST is compelling since a few distance measures and data sets are easily optimized [121]. Moreover, a unique validation index fed by biological theory is developed to be used for guiding many clustering approaches as well as B-MST.

B-MST is tested using 14 different data sets. Twelve of the data sets are used for external validation by the ARI measure. ARI values generated by K-means, and PAM are compared with values by B-MST. B-MST outperforms the other methods for 6 data sets. B-MST's performance is also compared with a well cited community structure algorithm, CSF. B-MST's highest rand index values are compared with CSF's values, since CSF is independent of distance measures. B-MST outperforms CSF in all of the data sets except Leukemia. The remaining two of the 14 data sets are used for biological inference. B-MST finds clusters with higher selectivity values than CSF, except for one cluster for the Yeast2 data set. B-MST finds biological process relevance in 7 clusters out of 8 while CSF finds

relevance in 6 of them. B-MST also finds higher selectivity values in most of the clusters than PAM for Yeast3. B-MST finds biological process relevance in 13 clusters while PAM finds in 10 out of 15.

The new TSI measure serves as a new quality measure to validate a result from a clustering algorithm using biological data. In external validation, minimum TSI values corresponds to maximum rand index values in most of the cases. In biological inference, CSF finds quite smaller TSI value for Yeast2, (100) compared to B-MST (100). B-MST finds a smaller TSI value, (845) compared to PAM (853). Hence, regarding biologically enriched genes in clusters with lower TSI values, the TSI is a good quality measure to be used in clustering biological data.

CHAPTER 3

CONCLUSION AND FUTURE RESEARCH

Clustering of high throughput biological data is a powerful method to guide biological experiments which impose high laboratory cost otherwise. Although many clustering algorithms exist, they are either general purpose or inefficient to handle high dimensional data. It is necessary to build efficient and effective algorithms that consider biological facts as much as possible. Here a MST based heuristic is developed and a new objective function is defined to assess the quality of partitions generated by the heuristic. The objective function uses transitive distances rather than pairwise which is biologically reasonable since an output is by the interactions of many biological components rather than two.

Different network topologies will affect the TSI value since TSI uses a binary network. For example, if the network is dense, clusters will have small diameters and the genes inside clusters will have a large number of connections with other clusters. Hence, B-MST and TSI measure should be used especially when the binary network is sparse. Another issue when the binary network is dense is that k_{max}^{out} will dominate the effect of D_{max}. This fact leads to parameter optimization study employing different co-efficients for both D_{max} and k_{max}^{out} parameters.

The optimization paradigm helps design powerful algorithm since clustering could be viewed as an optimization problem. The objective function is minimized using two

variables, D_{max} and k_{max}^{out}. The first one is for obtaining tight clusters and the latter one is for separating the clusters well.

Regarding the high dimensional nature of the gene expression data, a heuristic is developed and tested by comparing two commonly used and one recent and well cited clustering algorithms using 14 different data sets and 15 scenarios. Both external and biological validation indicate that the proposed method is both efficient and effective for clustering high throughput biological data.

For a future algorithmic study a mixed integer programming clustering model is developed as follows:

$$\text{Minimize} \quad D_{max} + k_{max}^{out}$$

subject to

$$\begin{aligned}
D_{max} &\geq d_{ij}(x_{is} + x_{js} - 1) && \forall i, j, s \;\; i < j && (3.1)\\
\sum_{s=1}^{c} x_{is} &= 1 && \forall i && (3.2)\\
\sum_{i=1}^{n} x_{is} &\geq 1 && \forall s && (3.3)\\
\sum_{j=1}^{n} A_{ij} x_{js} &\geq x_{is}\left(\frac{\sum_{j=1}^{n} A_{ij}}{2}\right) && \forall i, s && (3.4)\\
\sum_{j=1}^{n} A_{ij} x_{js} &\geq x_{is}\left(\sum_{j=1}^{n} A_{ij} - k_{max}^{out}\right) && \forall i, s && (3.5)\\
x_{is} &\in \{0, 1\} && \forall i, s && (3.6)\\
k_{max}^{out} &\geq 0 && && (3.7)
\end{aligned}$$

Model parameters are: n number of genes, c number of clusters, d_{ij} shortest path distances between genes i and j, A_{ij} is the adjacency of genes i and j , 1 if they are connected, 0 otherwise. Model variables are: x_{is} which are 1 if gene i is assigned to cluster s, 0 otherwise. D_{max} is the length of the largest diameter among all. k_{max}^{out} is the out connection number of the gene which has the maximum number of connections with the genes outside its cluster. 3.1 is the maximum diameter constraint. 3.2 ensures that each gene is assigned to exactly one cluster. 3.3 ensures that a cluster has at least one gene. 3.4 ensures that a gene has at least as many connections with genes inside its cluster as the number of connections with genes outside its cluster. 3.5 establishes the relation with objective function. 3.6 and 3.7 ensure that x_{is} are binary and k_{max}^{out} is real, greater than 0.

At the beginning of this research, first the model was developed. The model was solved using small data sets and two social networks data sets. The results led to the development of a heuristic to solve the model because of the high dimensional nature of biological data. However, since the model had tight constraints, such as the number of connections of a gene inside its clusters should be at least equal to the number of connections with other clusters, B-MST emerged independently from the model. Comparing this MIP model with a traditional clustering model and developing algorithms guided by the model are intended for future studies. Both the model and algorithms can be applied to relational data in fields such as biology and sociology.

REFERENCES

[1] Agarwal, G., Kempe, D., 2008. Modularity-maximizing graph communities via mathematical programming. The European Physical Journal B 66, 409–418.

[2] Al-Shahrour, F., Díz-Uriarte, R., Dopazo, J., 2004. Fatigo: a web tool for finding significant associations of gene ontology terms with groups of genes. Bioinformatics 20 (4), 578–580.

[3] Alderson, D. L., 2008. Catching the networkscience bug: insight and opportunity for the operations researcher. Operations Research 56 (5), 1047–1065.

[4] Allison, D. B., Page, G. P., Beasley, T. M., Edwards, J. W., 2005. DNA Microarrays and Related Genomics Techniques: Design, Analysis, and Interpretation of Experiments (Biostatistics). Chapman and Hall/CRC.

[5] Alshalalfah, M., Alhajj, R., 2009. Cancer class prediction: two stage clustering approach to identify informative genes. Intelligent Data Analysis 13 (4), 671–686.

[6] Andreopoulos, B., An, A., Wang, X., Schroeder, M., 2009. A roadmap of clustering algorithms: finding a match for a biomedical application. Briefings in Bioinformatics 10 (3), 297–314.

[7] Androulakis, I. P., 2009. Mathematical programming approaches for the analysis of microarray data. In: Handbook of Optimization in Medicine, Springer. Vol. 26. pp. 357 – 378.

[8] Arora, S., Rao, S., , Vazirani, U., 2004. Expander flows, geometric embeddings and graph partitioning. In: Proceedings of the thirty- sixth annual ACM symposium on Theory of computing. Chicago, pp. 222–231.

[9] Asyali, M. H., Colak, D., Demirkaya, O., Inan, M. S., 2006. Gene expression profile classification: a review. Current Bioinformatics 1, 55–73.

[10] Balasubramaniyan, R., Hüllermeier, E., Weskamp, N., Kämper, J., 2005. Clustering of gene expression data using a local shape-based similarity measure. Bioinformatics 21 (7), 1069–1077.

[11] Bandyopadhyay, S., Bhattacharyya, M., 2009. Analyzing mirna co-expression networks to explore tf-mirna regulation. BMC Bioinformatics 10 (163), 1–16.

[12] Bandyopadhyay, S., Mukhopadhyay, A., Maulik, U., 2007. An improved algorithm for clustering gene expression data. Bioinformatics 23 (21), 2859–2865.

[13] Bandyopadhyay, S., Pal, S. K., 2007. Dynamic range-based distance measure for microarray expressions and a fast gene-ordering algorithm. IEEE Transactions on Systems, Man, and Cybernetics 37 (3), 742–749.

[14] Barthelemy, P., Brucher, F., Osswald, C., 2007. Combinatorial optimisation and hierarchical classications. Annals of Operations Research 153 (1), 179–214.

[15] Berretta, R., Mendes, A., Moscato, P., 2005. Integer programming models and algorithms for molecular classification of cancer from microarray data. In: In Estivill-Castro 27. pp. 361–370.

[16] Beyer, A., 2009. Network-based models in molecular biology. In: Dynamics on and of Complex Networks. pp. 35–56.

[17] Bohland, J. W., Bokil, H., Pathak, S. D., Lee, C. K., Ng, L., Lau, C., Kuan, C., Hawrylycz, M., Mitra, P. P., 2010. Clustering of spatial gene expression patterns in the mouse brain and comparison with classical neuroanatomy. Methods 50 (2), 105–112.

[18] Boscolo, R., Sabatti, C., Liao, J. C., Roychowdhury, V. P., 2005. A generalized framework for network component analysis. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2 (4), 289–301.

[19] Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., Wagner, D., 2008. On modularity clustering. IEEE Transactions on Knowledge and Data Engineering 20 (2), 172–188.

[20] Bushel, P. R., 2009. Clustering of gene expression data and end-point measurements by simulated annealing. Journal of Bioinformatics and Computational Biology 7 (1), 193–215.

[21] Busygin, S., Prokopyev, O., Pardalos, P. M., 2008. Biclustering in data mining. Computers and Operations Research 35 (9), 2964–2987.

[22] Cano, C., Garcia, F., Lopez, F. J., Blanco, A., 2009. Intelligent system for the analysis of microarray data using principal components and estimation of distribution algorithms. Expert Systems with Applications 36 (3), 4654–4663.

[23] Ceccarelli, M., Maratea, A., 2008. Improving fuzzy clustering of biological data by metric learning with side information. International Journal of Approximate Reasoning 47 (1), 45–57.

[24] Chen, J., Hsu, W., Lee, M. L., Ng, S. K., 2005. Discovering reliable protein interactions from high-throughput experimental data using network topology. Artificial Intelligence in Medicine 35, 37–47.

[25] Chen, W. Y. C., Dress, A. W. M., Yu, W. Q., 2008. Community structure of networks. Mathematics in Computer Science 1 (3), 441–457.

[26] Chipman, H., Tibshirani, R., 2006. Hybrid hierarchical clustering with applications to microarray data. Biostatistics 7 (2), 286–301.

[27] Chouakria, A. D., Diallo, A., Giroud, F., 2009. Adaptive clustering for time series: application for identifying cell cycle expressed genes. Computational Statistics and Data Analysis 53 (4), 1414–1426.

[28] Christinat, Y., Wachmann, B., Zhang, L., 2008. Gene expression data analysis using a novel approach to biclustering combining discrete and continuous data. IEEE/ACM Transactions on Computational Biology and Bioinformatics 5 (4), 583–593.

[29] Clauset, A., Moore, C., , Newman, M. E. J., 2008. Hierarchical structure and the prediction of missing links in networks. Nature 453 (7191), 98–101.

[30] Clauset, A., Newman, M. E. J., Moore, C., 2004. Finding community structure in very large networks. Physical Review E 70, 1–6.

[31] Cohen, D. D., Kasif, S., Melkman, A. A., 2009. Seeing the forest for the trees: using the gene ontology to restructure hierarchical clustering. Bioinformatics 25 (14), 1789–1795.

[32] Csardi, G., Nepusz, T., 2006. The igraph software package for complex network research. InterJournal Complex Systems, 1695.

[33] D'haeseleer, P., 2005. How does gene expression clustering work? Nature Biotechnology 23 (12), 1499–1501.

[34] Dharan, S., Nair, A. S., 2009. Biclustering of gene expression data using reactive greedy randomized adaptive search procedure. BMC Bioinformatics 10 (S27), 1–10.

[35] Dittrich, M. T., Klau, G. W., Rosenwald, A., Dandekar, T., Müller, T., 2008. Identifying functional modules in protein-protein interaction networks: an integrated exact approach. Bioinformatics 24 (13), 223–231.

[36] Do, J. H., Choi, D. K., 2007. Clustering approaches to identifying gene expression patterns from dna microarray data. Molecules and Cells 25 (2), 279–288.

[37] Du, Z., Wang, Y., Ji, Z., 2008. Pk-means: a new algorithm for gene clustering. Computational Biology and Chemistry 32 (4), 243–247.

[38] Eckman, B. A., Brown, P. G., 2006. Graph data management for molecular and cell biology. IBM J. Res. and Dev. 50 (6), 545–560.

[39] Eisen, M. B., Spellman, P. T., Brown, P. O., Botstein, D., 1998. Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences of the United States of America 95 (25), 14863–14868.

[40] Ernst, J. ., Nau, G. J., Joseph, Z. B., 2005. Clustering short time series gene expression data. Bioinformatics 21 (1), 159–168.

[41] Faceli, K., Souto, M. C. P. D., Araujo, D. S. A. D., Carvalhoüç, A. C. P. L. F. D., 2009. Multi-objective clustering ensemble for gene expression data analysis. Neurocomputing 72, 2763–2774.

[42] Famili, A. F., Liu, G., Liu, Z., 2004. Evaluation and optimization of clustering in gene expression data analysis. Bioinformatics 20 (10), 1535–1545.

[43] Fathian, M., Amiri, B., Maroosi, A., 2007. Application of honey-bee mating optimization algorithm on clustering. Applied Mathematics and Computation 190 (2), 1502–1513.

[44] Fortunato, S., 2010. Community detection in graphs. Physics Reports 486, 75–174.

[45] Fujita, A., Sato, J. R., Demasi, M. A. A., Sogayar, M. C., 2009. Comparing pearson, spearman and hoeffding's d measure for gene expression association analysis. Journal of Bioinformatics and Computational Biology 7 (4), 663–684.

[46] Garge, N. R., Page, G. P., Sprague, A. P., Gorman, B. S., Allison, D. B., 2005. Reproducible clusters from microarray research: whither? BMC Bioinformatics 6 (S10), 1–11.

[47] Geraci, F., Leoncini, M., Montangero, M., Pellegrini, M., Renda, M. E., 2009. K-boost: a scalable algorithm for high-quality clustering of microarray gene expression data. Journal of Computational Biology 16 (6), 859–873.

[48] Ghouila, A., Yahia, S. B., Malouche, D., Jmel, H., Laouini, D., Guerfali, F. Z., Abdelhak, S., 2009. Application of multi-som clustering approach to macrophage gene expression analysis. Infection, Genetics and Evolution 9 (3), 328–336.

[49] Girvan, M., Newman, M. E. J., 2002. Community structure in social and biological networks. PNAS 99 (12), 7821–7826.

[50] Glover, F. W., Kochenberger, G., 2006. New optimization models for data mining. International Journal of Information Technology and Decision Making 5 (4), 605–609.

[51] Gómez, S., Jensen, P., Arenas, A., 2009. Analysis of community structure in networks of correlated data. Physical Review E 80 (016114), 1–5.

[52] Gungor, Z., Unler, A., 2007. K-harmonic means data clustering with simulated annealing heuristic. Applied Mathematics and Computation 184 (2), 199–209.

[53] Gungor, Z., Unler, A., 2008. K-harmonic means data clustering with tabu-search method. Applied Mathematical Modelling 32 (6), 1115–1125.

[54] Hagberg, A. A., Schult, D. A., Swart, P. J., Aug. 2008. Exploring network structure, dynamics, and function using NetworkX. In: Proceedings of the 7th Python in Science Conference (SciPy2008). Pasadena, CA USA, pp. 11–15.

[55] Hageman, J. A., Berg, R. A. V. D., Westerhuis, J. A., Werf, M. J. V. D., Smilde, A. K., 2008. Genetic algorithm based two-mode clustering of metabolomics data. Metabolomics 4 (2), 141–149.

[56] Hayashida, M., Sun, F., Aburatani, S., Horimoto, K., Akutsu, T., 2007. Integer programming-based approach to allocation of reporter genes for cell array analysis. In: The First International Symposium on Optimization and SystemsBiology(OSB07). pp. 288–301.

[57] He, Y., Hui, S. C., 2009. Exploring ant-based algorithms for gene expression data analysis. Artificial Intelligence in Medicine 47 (2), 105–119.

[58] Heath, J. W., Fu, M. C., Jank, W., 2009. New global optimization algorithms for model-based clustering. Computational Statistics and Data Analysis 53 (12), 3999–4017.

[59] Higham, D. J., Kalna, G., 2007. Spectral analysis of two-signed microarray gene expression data. Mathematical Medicine and Biology 24 (2), 131–148.

[60] Higham, D. J., Kalna, G., Kibble, M., 2007. Spectral clustering and its use in bioinformatics. Journal of computational and applied mathematics 204, 25–37.

[61] Hilt, S. W., Yelundur, A., McChesney, C., Landry, M., 2006. Support vector machine implementations for classification and clustering. BMC Bioinformatics 7 (S4), 1–18.

[62] Horst, E., 2003. Distance measures for mpeg-7-based retrieval. In: MIR '03: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval. ACM, New York, NY, USA, pp. 130–137.

[63] Hoshida, Y., Brunet, J.-P., Tamayo, P., Golub, T. R., Mesirov, J. P., 11 2007. Subclass mapping: Identifying common subtypes in independent disease data sets. PLoS ONE 2 (11), e1195.

[64] Hu, X., Ng, M., Wu, F. X., Sokhansanj, B. A., 2009. Mining, modeling, and evaluation of subnetworks from large biomolecular networks and its comparison study. IEEE Transactions on Information Technology in Biomedicine 13 (2), 184–194.

[65] Hu, X., Park, E. K., Zhang, X., 2009. Microarray gene cluster identication and annotation through cluster ensemble and em-based informative textual summarization. IEEE Transactions on Information Technology in Biomedicine 13 (5), 832–840.

[66] Huang, D., Pan, W., 2006. Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. Bioinformatics 22 (10), 1259–1268.

[67] Hubert, L., Arabie, P., 1985. Comparing partitions. Journal of Classification 2, 193–218.

[68] Huttenhower, C., Flamholz, A. I., Landis, J. N., Sahi, S., Myers, C. L., Olszewski, K. L., Hibbs, M. A., Siemers, N. O., Troyanskaya, O. G., Coller, H. A., 2007. Nearest neighbor networks: clustering expression data based on gene neighborhoods. BMC Bioinformatics 8 (250), 1–13.

[69] Iyer, V. R., Eisen, M. B., Ross, D. T., Schuler, G., Moore, T., Lee, J. C., Trent, J. M., Staudt, L. M., Hudson, J., Boguski, M. S., Lashkari, D., Shalon, D., Botstein, D., Brown, P. O., 1999. The transcriptional program in the response of human fibroblasts to serum. Science 283 (5398), 83–87.

[70] Jarboui, B., Cheikh, M., Siarry, P., Rebai, A., 2007. Combinatorial particle swarm optimization (cpso) for partitional clustering problem. Applied Mathematics and Computation 192 (2), 337–345.

[71] Ji, X., 2004. Graph partition problems with minimum size constraints. Ph.D. thesis, Rensselaer Polytechnique Institute.

[72] Jiang, D., Tang, C., Zhang, A., 2004. Cluster analysis for gene expression data: a survey. IEEE Transactions on Knowledge and Data Engineering 16 (11), 1370–1386.

[73] Jonnalagadda, S., Srinivasan, R., 2009. Nifti: An evolutionary approach for finding number of clusters in microarray data. BMC Bioinformatics 10 (40), 1–13.

[74] Joshi, A., Smet, R. D., Marchal, K., Peer, Y. V. D., Michoel, T., 2009. Module networks revisited: computational assessment and prioritization of model predictions. Bioinformatics 25 (4), 490–496.

[75] Kanehisa, M., Bork, P., 2003. Bioinformatics in the post-sequence era. Nature Genetics 33, 305–310.

[76] Kaufman, L., Rousseeuw, P., 1990. Finding groups in data: an introduction to cluster analysis. Wiley and Sons.

[77] Kelley, L. A., Gardner, S. P., Sutcliffe, M. J., 1996. An automated approach for clustering an ensemble of nmr- derived protein structures into conformationally related subfamilies. Protein Engineering 9 (11), 1063–1065.

[78] Kernighan, B. W., Lin, S., 1970. An efficient heuristic procedure for partitioning graphs,. Bell System Technical Journal 49 (2), 291–308.

[79] Kim, J., Choi, S., 2006. Semidefinite spectral clustering. Pattern Recognition 39, 2025–2035.

[80] Korkmaz, E. E., Du, J., Alhajj, R., , Barker, K., 2006. Combining advantages of new chromosome representation scheme and multi-objective genetic algorithms for better clustering. Intelligent Data Analysis 10 (2), 163–182.

[81] Lancichinetti, A., Radicchi, F., 2008. Benchmark graphs for testing community detection algorithms. Physical Review E 78 (4).

[82] Langfelder, P., Zhang, B., Horvath, S., 2008. Defining clusters from a hierarchical cluster tree: the dynamic tree cut package for r. Bioinformatics Applications Note 24 (5), 719–720.

[83] Lau, J. W., Green, P. J., 2007. Bayesian model-based clustering procedures. Journal of Computational and Graphical Statistic 16 (3), 526–558.

[84] Lee, C. H., Zaiane, O. R., Park, H. H., Huang, J., Greiner, R., 2008. Clustering high dimensional data: A graph-based relaxed optimization approach. Information Sciences 178 (23), 4501–4511.

[85] Lee, W. P., Tzou, W. S., 2009. Computational methods for discovering gene networks from expression data. Briefings in Bioinformatics 10 (4), 408–423.

[86] Li, G., Ma, Q., Tang, H., Paterson, A. H., Xu, Y., 2009. Qubic: a qualitative biclustering algorithm for analyses of gene expression data. Nucleic Acids Research 37 (15), 1–10.

[87] Li, J., Halgamuge, S. K., Tang, S. L., 2008. Genome classification by gene distribution: An overlapping subspace clustering approach. BMC Evolutionary Biology 8 (116), 1–15.

[88] Liang, F., Wang, N., 2007. Dynamic agglomerative clustering of gene expression proles. Pattern Recognition Letters 28 (9), 1062–1076.

[89] Liu, J., Li, Z., Hu, X., Chen, Y., 2009. Biclustering of microarray data with mospo based on crowding distance. BMC Bioinformatics 10 (S9), 1–10.

[90] Liu, T., Lin, N., Shi, N., Zhang, B., 2009. Information criterion-based clustering with order-restricted candidate profiles in short time-course microarray experiments. BMC Bioinformatics 10 (146), 1–20.

[91] Liu, X., Lee, S. C., Casella, G., Peter, G. F., 2008. Assessing agreement of clustering methods with gene expression microarray data. Computational Statistics and Data Analysis 52 (12), 5356–5366.

[92] Lucchetti, R., Moretti, S., Patrone, F., Radrizzani, P., 2009. The shapley and banzhaf values in microarray games. Computers and Operations Research CAOR 2342.

[93] Ma, P. C. H., Chan, K. C. C., 2009. An iterative data mining approach for mining overlapping co-expression patterns in noisy gene expression data. IEEE Transactions on NanoBioscience 8 (3), 252–258.

[94] Ma, P. C. H., Chan, K. C. C., 2009. A novel approach for discovering overlapping clusters in gene expression data. IEEE Transactions on Biomedical Engineering 56 (7), 1803–1808.

[95] Madeira, S. C., Oliveira, A. L., 2004. Biclustering algorithms for biological data analysis: a survey. IEEE Transactions on Computational Biology and Bioinformatics 1 (1), 24–45.

[96] Manning, C. D., Raghavan, P., Schutze, H., 2009. An introduction to information retrieval. Cambridge University Press Online Edition.

[97] Marketa, Z., Jeremy, O. B., 2008. Understanding bioinformatics. Garland Science.

[98] Maulik, U., Mukhopadhyay, A., 2010. Simulated annealing based automatic fuzzy clustering combined with ann classification for analyzing microarray data. Computers and Operations Research 37 (8), 1369–1380.

[99] McAllister, S. R., DiMaggio, P. A., Floudas, C. A., 2009. Mathematical modeling and efficient optimization methods for the distance-dependent rearrangement clustering problem. J. Glob. Optim. 45 (1), 111–129.

[100] Melia, M., Pentney, W., 2007. Clustering by weighted cuts in directed graphs. Proceedings of SIAM Conference on Data Mining.

[101] Meng, J., Gao, S. J., Huang, Y., 2009. Enrichment constrained time-dependent clustering analysis for nding meaningful temporal transcription modules. Bioinformatics 25 (12), 1521–1527.

[102] Merz, P., 2003. Analysis of gene expression profiles: an application of memetic algorithms to the minimum sum-of-squares clustering problem. BioSystems 72 (1-2), 99–109.

[103] Mete, M., Tang, F., Xu, X., Yuruk, N., 2008. A structural approach for finding functional modules from large biological networks. BMC Bioinformatics 9 (S19), 1–14.

[104] Mitra, S., Das, R., Banka, H., Mukhopadhyay, S., 2009. Gene interaction - an evolutionary biclustering approach. Information Fusion 10, 242–249.

[105] Monti, S., Tamayo, P., Mesirov, J., Golub, T., 2003. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning 52 (1-2), 91–118.

[106] Moretti, S., 2009. Statistical analysis of the shapley value for microarraygames. Computers and Operations Research CAOR 2341.

[107] Mukhopadhyay, A., Maulik, U., 2009. Towards improving fuzzy clustering using support vector machine: Application to gene expression data. Pattern Recognition 42 (11), 2744–2763.

[108] Nascimento, M. C. V., Toledo, F. M. B., Carvalho, A. C. P. L. F. D., 2010. Investigation of a grasp-based clustering algorithm applied to biological data. Computers and Operations Research 37 (8), 1381–1388.

[109] Newman, M. E. J., 2004. Analysis of weighted networks. Physical Review E 70 (056131), 1–9.

[110] Newman, M. E. J., 2004. Detecting community structure in networks. The European Physical Journal B 38 (2), 321–330.

[111] Newman, M. E. J., 2006. Finding community structure in networks using the eigen vectors of matrices. Physical Review E 74 (036104).

[112] Newman, M. E. J., 2006. Modularity and community structure in networks. PNAS 103 (23), 8577–8582.

[113] Newman, M. E. J., Girvan, M., 2004. Finding and evaluating community structure in networks. Physical Review E 69 (026113), 1–15.

[114] Nueda, M. J., Sebastián, P., Tarazona, S., García, F. G., Dopazo, J., Ferrer, A., Conesa, A., 2009. Functional assessment of time course microarray data. BMC Bioinformatics 10 (S9), 1–18.

[115] Palla, G., Derényi, I., Farkas, I., Vicsek, T., 2005. Uncovering the overlapping community structure of complex networks in nature and society. Nature 435 (9), 814–818.

[116] Phan, V., George, E. O., Tran, Q. T., Goodwin, S., 2009. Analyzing microarray data with transitive directed acyclic graphs. Journal of Bioinformatics and Computational Biology 7 (1), 135–156.

[117] Qin, Z. S., 2006. Clustering microarray gene expression data using weighted chinese restaurant process. Bioinformatics 22 (16), 1988–1997.

[118] Ravi, V., Bin, M., Kumar, P. R., 2006. Threshold accepting based fuzzy clustering algorithms. International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems 14 (5), 617–632.

[119] Richards, A. L., Holmans, P., O'Donovan, M. C., Owen, M. J., Jones, L., 2008. A comparison of four clustering methods for brain expression microarray data. BMC Bioinformatics 9 (490), 1–17.

[120] Robbins, K. R., Zhang, W., Bertrand, J. K., 2007. The ant colony algorithm for feature selection in high-dimension gene expression data for disease classification. Mathematical Medicine and Biology 24 (4), 413–426.

[121] Rocke, D. M., Ideker, T., Troyanskaya, O., Quackenbush, J., Dopazo, J., 2009. Papers on normalization, variable selection, classification or clustering of microarray data. Bioinformatics 25 (6), 701–702.

[122] Rousseeuw, P. J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53–65.

[123] Ruan, J., Zhang, W., 2008. Identifying network communities with a high resolution. Physical Review E 77 (016104), 1–12.

[124] Saha, S., Bandyopadhyay, S., 2009. A new point symmetry based fuzzy genetic clustering technique for automatic evolution of clusters. Information Sciences 179 (19), 3230–3246.

[125] Scharl, T., Leisch, F., 2009. gcexplorer: interactive exploration of gene clusters. Bioinformatics 25 (8), 1089–1090.

[126] Scharl, T., Voglhuber, I., Leisch, F., 2009. Exploratory and inferential analysis of gene cluster neighborhood graphs. BMC Bioinformatics 10 (288), 1–14.

[127] Schwarz, A. J., Gozzi, A., Bifone, A., 2009. Community structure in networks of functional connectivity: resolving functional organization in the rat brain with pharmacological mri. NeuroImage 47 (1), 302–311.

[128] Shaik, Z. S., Yeasin, M., 2007. A unified framework for finding differentially expressed genes from microarray experiments. BMC Bioinformatics 8 (347), 1–21.

[129] Sharma, A., Podolsky, R., Zhao, J., McIndoe, R. A., 2009. A modied hyperplane clustering algorithm allows for efcient and accurate clustering of extremely large datasets. Bioinformatics 25 (9), 1152–1157.

[130] Shen, R., Olshen, A. B., Ladanyi, M., 2009. Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. Bioinformatics 25 (22), 2906–2912.

[131] Steggies, L. J., Banks, R., Shaw, O., Wipat, A., 2007. Qualitatively modeling and analyzing genetic regulatory networks: a petri net approach. Bioinformatics 23 (3), 336–343.

[132] Stone, E. A., Ayroles, J. F., 2009. Modulated modularity clustering as an exploratory tool for functional genomic inference. PLoS Genetics 5 (5), 1–13.

[133] Tan, M. P., Broach, J. R., Floudas, C. A., 2007. Evaluation of normalization and pre-clustering issues in a noval clustering approach: global optimum search with enhanced positioning. Journal of Bioinformatics and Computational Biology 5 (4), 895–913.

[134] Tan, M. P., Broach, J. R., Floudas, C. A., 2007. A novel clustering approach and prediction of optimal number of clusters: global optimum search with enhanced positioning. Journal of Global Optimization 39 (3), 323–346.

[135] Tan, M. P., Smith, E. N., Broach, J. R., Floudas, C. A., 2008. Microarray data mining: A novel optimization-based approach to uncover biologically coherent structures. BMC Bioinformatics 9 (268), 1–21.

[136] Teboulle, M., 2007. A unied continuous optimization framework for center-based clustering methods. Journal of Machine Learning Research 8, 65–102.

[137] Thalamuthu, A., Mukhopadhyay, I., Zheng, X., Tseng, G. C., 2006. Evaluation and comparison of gene clustering methods in microarray analysis. Bioinformatics 22 (19), 2405–2412.

[138] Tibely, G., Kertesz, J., 2008. On the equivalence of the label propagation method of community detection and a potts model approach. Physica A: Statistical Mechanics and its Applications 387 (19-20), 4982–4984.

[139] Torrente, A., Kapushesky, M., Brazma, A., 2005. A new algorithm for comparing and visualizing relationships between hierarchical and at gene expression data clusterings. Bioinformatics 21 (21), 3993–3999.

[140] Tritchler, D., Parkhomenko, E., Beyene, J., 2009. Filtering genes for cluster and network analysis. BMC Bioinformatics 10 (193), 1–9.

[141] Tseng, G. C., 2007. Penalized and weighted k-means for clustering with scattered objects and prior information in high-throughput biological data. Bioinformatics 23 (17), 2247–2255.

[142] Tseng, G. C., Wong, W. H., 2005. Tight clustering: a resampling-based approach for identifying stable and tight patterns in data. Biometrics 61 (1), 10–16.

[143] Tu, Y., Stolovitzky, G., Klein, U., 2002. Quantitative noise analysis for gene expression microarray experiments. PNAS 99 (22), 1403114036.

[144] Tyler, A. L., Asselbergs, F. W., Williams, S. M., Moore, J. H., 2009. Shadows of complexity: what biological networks reveal about epistasis and pleiotropy. BioEssays 31 (2), 220–227.

[145] Wang, K., Zheng, J., Zhang, J., Dong, J., 2009. Estimating the number of clusters via system evolution for cluster analysis of gene expression data. IEEE Transactions on Information Technology in Biomedicine 13 (5), 848–853.

[146] Wang, S., Zhu, J., 2008. Variable selection for model-based high-dimensional clustering and its application to microarray data. Biometrics 64 (2), 440–448.

[147] Wei, L. Y., Cheng, C. H., 2008. An entropy clustering analysis based on genetic algorithm. Journal of Intelligent and Fuzzy Systems 19 (4-5), 235–241.

[148] Wild, D. J., Blankley, C. J., 2000. Comparison of 2d fingerprint types and hierarchy level selection methods for structural grouping using wards clustering. J. Chem. Inf. Comput. Sci. 40, 155–162.

[149] Wu, F. X., 2008. Genetic weighted k-means algorithm for clustering large-scale gene expression data. BMC Bioinformatics 9 (S12), 1–10.

[150] Xie, B., Pan, W., Shen, X., 2008. Variable selection in penalized model-based clustering via regularization on grouped parameters. Biometrics 64 (3), 921–930.

[151] Xu, Y., Olman, V., Xu, D., 2002. Clustering gene expression data using graph-theoretic approach: an application of minimum spanning trees. Bioinformatics 18 (4), 536–545.

[152] Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E., Ruzzo, W. L., 2001. Model-based clustering and data transformations for gene expression data. Tech. rep., UW-CSE.

[153] Yeung, K. Y., Haynor, D. R., Ruzzo, W. L., 2001. Validating clustering for gene expression data. Bioinformatics 17 (4), 309–318.

[154] Yi, Q., Shizhong, X., 2004. Supervised cluster analysis for microarray data based on multivariate gaussian mixture. Bioinformatics 20 (12), 1905–1913.

[155] Yip, A. M., Ng, M. K., Wu, E. H., Chan, T. F., 2007. Strategies for identifying statistically significant dense regions in microarray data. IEEE/ACM Transactions on Computational Biology and Bioinformatics 4 (3), 415–429.

[156] Yu, Z., Wong, H. S., 2009. Class discovery from gene expression data based on perturbation and cluster ensemble. IEEE Transactions on Nanobioscience 8 (2), 147–160.

[157] Yujin, H., Philippe, B. J., Pablo, T., R., G. T., P., M. J., 11 2007. Subclass mapping: identifying common subtypes in independent disease data sets. PLoS ONE 2 (11).

[158] Zahoránszky, L. A., Katona, G. Y., Hári, P., Csizmadia, A. M., Zweig, K. A., Köhalmi, G. Z., 2009. Breaking the hierarchy - a new cluster selection mechanism for hierarchical clustering methods. Algorithms for Molecular Biology 4 (12), 1–22.

[159] Zhang, B., Horvath, S., 2005. A general framework for weighted gene co-expression network analysis. Statistical Applications in Genetics and Molecular Biology, The Berkeley Electronic Press 4 (17).

[160] Zhang, W., Fang, H. B., Song, J., 2009. Principal component tests: applied to temporal gene expression data. BMC Bioinformatics 10 (S26), 1–9.

[161] Zhang, Y., Xuan, J., Reyes, B. G. D. L., Clarke, R., Ressom, H. W., 2009. Reverse engineering module networks by pso-rnn hybrid modeling. BMC Genomics 10 (S15), 1–10.

[162] Zhou, X., Kao, M. C. J., Wong, W. H., 2002. Transitive functional annotation by shortest-path analysis of gene expression data. PNAS 99 (20), 12783–12788.

[163] Zhu, D., Dequeeant, M. L., Li, H., 2008. Comparative analysis of clustering methods for microarray data. In: Analysis of Microarray Data, Wiley. pp. 27–50.

[164] Zhu, D., Hero, A. O., Cheng, H., Khanna, R., Swaroop, A., 2005. Network constrained clustering for gene microarray data. Bioinformatics 21 (21), 4014–4020.